blender art MAGAZINE

CAR MODELING

- Textured Metal Shaders
- Making A Low Poly Car
- Modeling A Car Rim
- Modeling Tires
- Making Of 'Cutting The Waves'
- Making Of Scale Model
- Car Body Modeling: An Approach

www.blenderart.org

Happy New Year!

Sandra Gilbert
Managing Editor

Go to almost any CG site on the web and you will find them. Cars; hundreds of CG cars can be found in every shape, size and color. Little cars, big cars, fancy cars, sports cars, toon cars, radical designs and even economical cars; they are everywhere. Cars have long been a popular subject for modelers. They take great skill and look amazing when done right. You could spend hours tweaking everything from chrome details to the treads on the tires. And a great many modelers do exactly that.

Many modelers not only model from blueprints or reference photos, but actually design their own cars. It's CG; they can add or delete any part they choose. Colors can be changed at will. Different settings can be created to showcase their new design. And with the advances in rendering engines, all too often the resulting images can be/are mistaken for actual photos of real cars. Many car manufacturers take advantage of this and provide us with excellent advertisements too.

Wouldn't it be great if you could actually build your concept car? But that is just wishful thinking ... or is it? Marin Myftiu shows how he built a prototype of a concept design car using Blender in combination with other techniques.

Also in this issue we cover tires, rims and a low poly car in a simple driving game. Gaurav is going to take us through some 'approaches in modeling a car body' via the Nissan 350z model. And as an added treat Claas Kuhnen shows us how to create his amazing brushed metal textures; useful for car interiors as well as for jewelry and a host of other metal needs.

So rev up your blender engine and let's get this party started!

Happy Blendering
sandra@blenderart.org

CONTENTS

EDITOR/DESIGNER
Gaurav Nawani
gaurav@blenderart.org

MANAGING EDITOR
Sandra Gilbert
sandra@blenderart.org

WEBSITE
Nam Pham
nam@blenderart.org

PROOFER
Kernon Dillon
Phillip A. Ryals
Kevin C. Braun
Derek Marsh

WRITERS
Claas Eicke Kuhnen
Zsolt Stefan
Marin Myftiu
Pablo Delgado
Thomas Baron
Sandra Gilbert
Gaurav Nawani

COVER ART
Philippe Roubal

Upcoming Release Alert

Blender Release

2006 has come and gone, bringing with it a large number of exciting changes and events. I had planned on doing a review of 2006, but b@rt beat me to it and did such a good job, there is no point in doing it again. If you missed it, you can check out his 2006 review of Blender at:

(http://www.blendernation.com/2006/12/31/looking-back-on-2006/).

The release of Blender 2.43 is on the horizon. RC1 and RC2 have already been released for bug testing and the official release should be following within a few weeks. The release has quite a few new features to play with. Of particular interest to me are the Walk-Cycle Modifiers.

The Walk-Cycle Modifiers replace the Stride bone allowing for an easier setup of walk cycles for use in the NLA. Additionally, an Action Modifier allows you to use a Curve Path to deform the motion of controlling bones in the armature. Following is a list of included new features:

Modeling

Modifier stack upgrades.
New modifiers: Edgesplit and Displace
UV Project, Sculpt modeling
Multi-Resolution Mesh and Retopo
Fluid Dynamics supporting animated Objects
Multi-level UV editing
Face Duplicators

Animation

Walk Cycle Modifiers
Proxy Objects, for local control over referenced data from Libraries

Rendering

Render features: Alpha Masks, Node Shader speedup, Tangents and Normal Maps
Irregular Shadow Buffers
Shadow Buffer, Halfway Average
Render Baking
Render Passes

Compositing

UV Map, Index Mask and Z-Combine
Defocus
Matte Nodes
Matte Nodes, tutorial

Imaging

Multi-layer images, Sequence images, flip books

For further updates and current release logs, go to:
http://www.blender.org/cms/Changes_since_2_42.771.0.html

2007 Blender F1

The 2007 Blender F1 contest is here again. CurtisS is running the contest again this year and will soon have the official contest site up and ready with all the rules and specifications for the images.

Check at the Blender F1 Contest site for last years results. Also, check the discussion thread in the forums, for updated news and to just generally discuss the contest.

Good luck and get ready to start your blender engines!!!!!!!!

How To Model Cars

Introduction

Modeling cars can be as easy or as complicated as you want to make it. You can design your own car or model existing cars. It is amazing how much information is available to help you get your car modeled. In fact, the forum database at www.blenderartists.org has pages of links to great tips, tutorials and images of some amazing cars.

Here are a few general tips to get you going.

General Steps to modeling a car
Step 1: Get blueprints or create sketches of your design concept.
Step 2: Set the blueprints up in the Top, Right, and Front view ports using View|Background Image.
Step 3: Add a Plane, position it near the center-line.
Step 4: Mirror the Plane, so you only have to model half of the car.
Step 5: Extrude the edges of the Plane and move vertices around to match the profile of the car in the Right view.
Step 5a: Move vertices around to match the profile of the car in the Top view.
Step 5b: Move vertices around to match the profile of the car in the Front view.
Step 6: Repeat Step 5 until the model is finished.
Step 7: Add some textures, work on lighting, and do a render.

To make it easier to model the interior, you can move the roof section to another position or Layer to make it easier to see inside the car.

It is easier to model the wheel wells instead of trying to cut them out later.

Seams on car

Cars have lots of seams and small gaps where parts fit together. And since subsurf is generally used on car models, seams get smoothed away. To create seams, you can crease the targeted edges with 'Shift + E' or, you can model 3-4 rows of vertices close together.

Painting your car

At some point, you will want to paint your car; whether it is a toony car or a super realistic one. It can be as complicated or as simple as you want to make it. Here are some resources to help get you started.

A demo car paint material, using nodes, can be found here (posted by "broken"): http://blenderartists.org/forum/showthread.php?t=71232&highlight=car+paint

He has also posted a video tutorial that shows you how to fake a layered car paint material

Sonix created a library of car paints/chrome/glass/tires and misc materials needed for cars:
Blender 2.34 Car Material Library Release 1.
http://www.free-webspace.biz/sonix/

Low Poly Car And Small Driving Gamelet

- Sandra Gilbert (dreamsgate)

Level: Beginer to Intermediate

Introduction

Low poly cars can be a lot of fun; both to make and to play with. You can make them as fancy or as simple as you want. I'm going to show you how to make a simple low poly car and then we are going to create a gamelet (gamelet = more than a demo, not quite a full fledged game) so we can play with our car.

I used Blender 2.43 RC1 during the writing of this tutorial. Therefore it should be compatible with Blender 2.43 when it releases. It should also be compatible with 2.42 as we are not going to use a lot of game features to set this up. This is an intermediate level tutorial although beginners should be able to follow along without much difficulty.

Low Poly Car

As with a great many models, we are going to start with a plane. We will model 1 half of the car and then mirror the other side.

1.In side view, add the car reference image View> Background Image> Load (load supplied reference image "Car")
2.Then add a Plane; Spacebar> Add> Plane, delete all but 1 vertice.
3.We are trying to make as few faces as possible so only add as many vertices as needed. Select the remaining vertice and Control LMB to create an outline of the car. Then create faces to fill in your outline by selecting 4 vertices and hitting "F key". (Fig 1)
4.In top view, select all the vertices and extrude (E key) 2 times. (Fig 2)
5.Select the outer row of vertices and hit Smooth a few times. (Fig 3))
6.Add a Mirror modifier, you may have to cycle through the X,Y,Z axis buttons to get it to mirror on the correct axis, also make sure your pivot point (little circle/dot in center of model) is positioned on the center line. And select "Do clipping" to get a clean join between the 2 halves.

Okay, there it is. One low poly sports

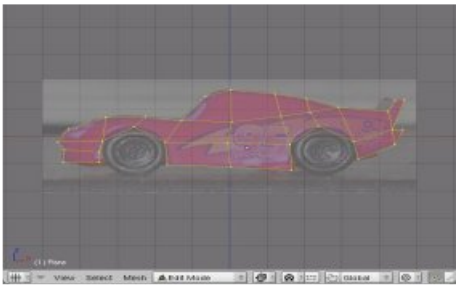


Fig 1: Lowpoly mesh.

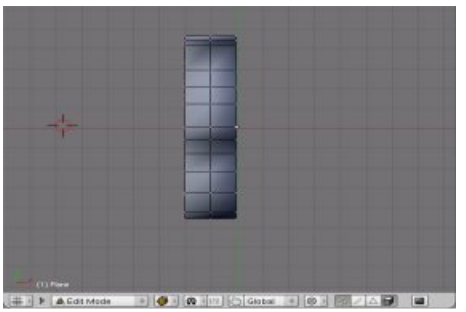


Fig 2: Extruding the car surface.

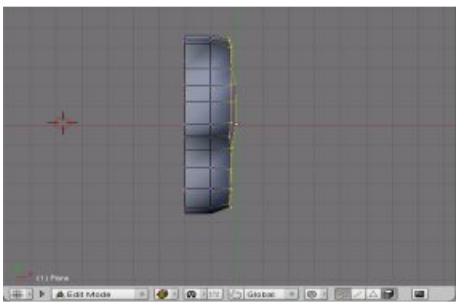


Fig 3: Applying smooth to the selection.

car. Now we need to add some wheels. They don't need to be fancy as they won't be really seen in our gamelet. So I just used a scaled tube for each wheel and joined it to the body. The reason I used a tube was that I wanted to close only one end and create a simple rim for

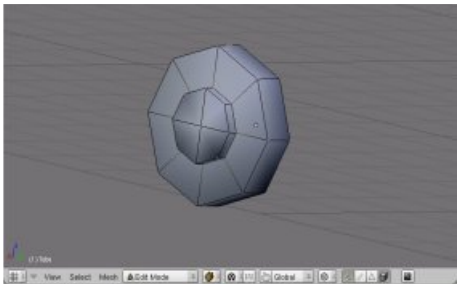


Fig 4: Lowpoly wheel.

painting. See fig 4
Next Duplicate your tire (Shift + D) and place both tires in their wheel wells. Select both tires and the car and Join (Control + J). See fig 5. I haven't applied (made final) the Mirror modifier yet because I want to vertex paint the car and I want both sides to be the same.

Painting the Car

At this point you can go off and apply a fancy UV map to your car or apply simple vertex paint. The choice is yours. For this tutorial, I decided to use vertex paint to keep

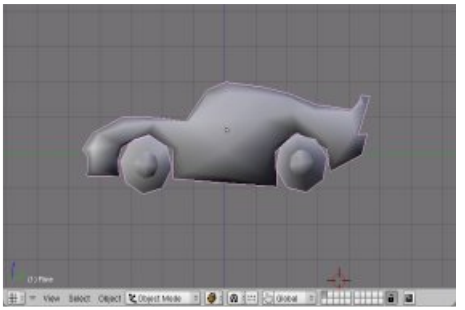


Fig 5: Ready car model.

memory requirements down. Remember to view your model in Potato mode (Alt + Z) to see your vertex paint.

1.Hit "V key" for Vertex paint and "F key" for UV/Face mode. Most likely your car turned completely black. That's ok; we will fix it in a moment.

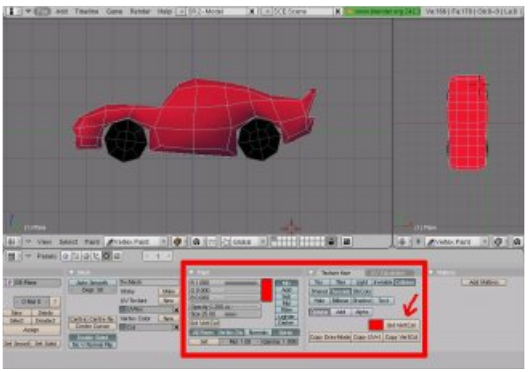


Fig 6: Vertex paint mode.

2.Select all the faces of the car (but not the tires) and in the Paint Panel in the Edit buttons window, set your paint color. I chose red. Hit the "Set Vert Color" in the Texture Face panel to apply your color to the car body. See fig 6.
3.I'm going to create some shadows and highlights using vertex paint, other wise if you press "P" to start the game engine you will notice that your car has no real definition and looks rather ugly.
4.Set your Brush size down to around 10-12 and opacity to 0.200. Choose a color darker than you chose for your car body. Start painting around the edges to give it definition. This is what I came up with. Not beautiful, but it's a start. See fig 7. Notice I also painted a little blue on the black tires and made the hub of the wheel a light grayish color. You could also add

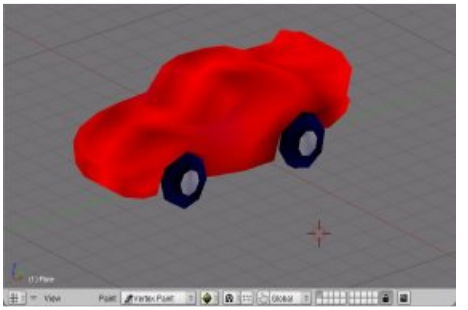


Fig 7: The car after painting.

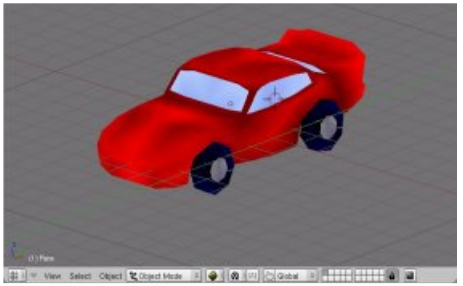


Fig 8: Mirrors painted with light blue.

windows and paint them a very light blue color. See fig 8.

Well our car is finished at this point. Now go ahead and apply the Mirror Modifier and make your car all one piece. Next we will be making a simple environment to drive around in.

Drive through the Country

We are going to make a driving course through the country for our car to drive around in.

1.Go to layer 2

2.In Top view, Spacebar>Add>Grid (default of 32 x 32 is fine)

3.In Edit mode, toggle the Face selection button and create a track for your car. See fig 9.

4.In Side view, Extrude your track down a little. See fig 10.

5.Switch back to Object mode (Tab), then go into UV/Face Mode (F) and Vertex Paint (V)

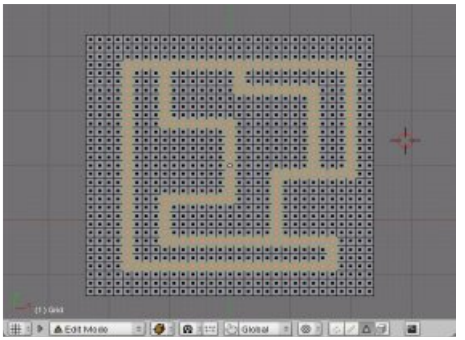


Fig 9: Selection for tracks.

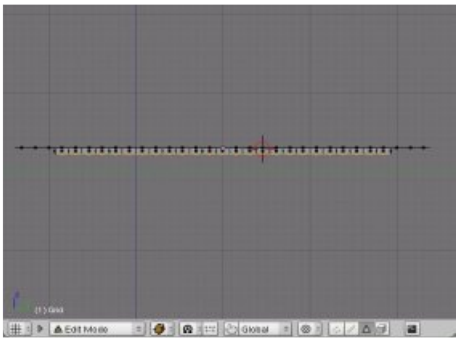


Fig 10: Extruding the tracks down.

6.Set your paint color to a muddy brown and press the “Set Vert Color button”, Tab back to Edit mode and use the “Select Swap button” in the Mesh tools 1 panel to invert your selection. Tab back to UV/Face mode and set your paint color to a nice green and press the “Set Vert Color button”. See fig 11.

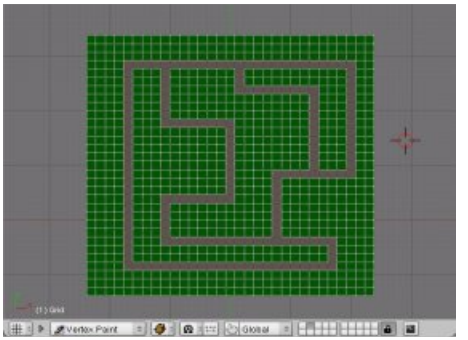


Fig 11: Painted tracks.

Now you have your basic track, but it is a little boring. I’m going to add a few hills and valleys using the Proportional edit tool (O). You have several choices of “falloff” to choose from, and I’ll leave it to you to decide which one you want. In fact mix and match them for different effects. See fig 12.

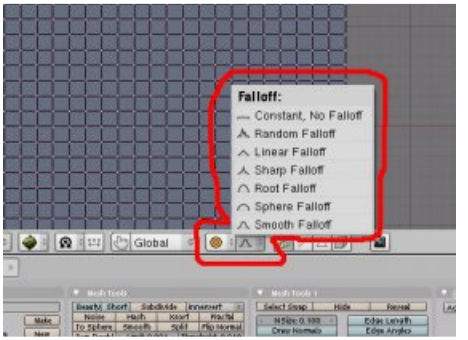


Fig 12: Proportional editing for variations.

1.For ease in selecting where I want my hills, I selected the faces I wanted in UV/Face select mode (so I could see where they were in relation to the track, you will notice your vertex colors don't show up in Edit mode).
2.Then Tab back to Edit Mode, with Proportional editing on (O) in side view, grab the faces (G) and move them up. They look a little too blocky to be real hills, so hit the Smooth button a few times, and then you may want to move them up a little bit more. See fig 13.

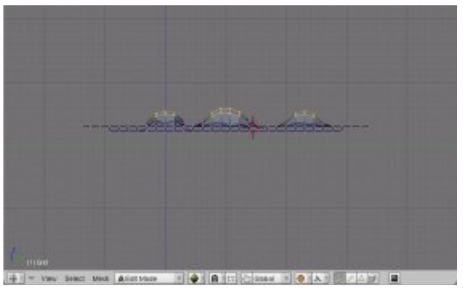


Fig 13: Scene with more variations.

3.Using the same technique, create valleys by moving down, not up. Also, don't be afraid to select portions of your track and move them up or down also. You can also use the Select>Random in the file menus. After playing around, this is what I got. I added some vertex paint to break up the flat greens and brown colors. See fig 14. (You might discover when you get your gamelet set up that your terrain is a little too rough, this can be solved by applying a Subsurf Modifier to smooth it out a little)

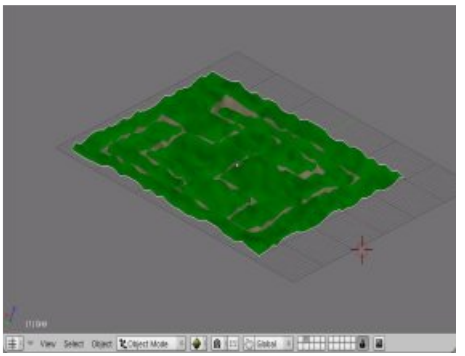


Fig 14: Smoothed surface.

Making a Gamelet

Okay, now we have a car and an environment for the car to drive around in. It is time to put together our little gamelet so we can play with it.

1.Turn on layer 1 so you can see your car. You will need to resize your car so that it fits on the track. After you have resized it, make sure that you hit "Control + A" to apply scale and rotation. It's a good idea to select the track/environment and do the same thing. See fig 15.

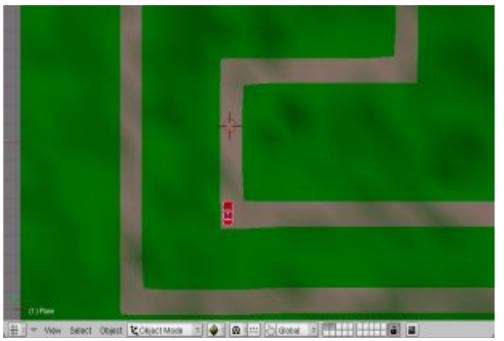


Fig 15: Resetting the car cordinates.

TIP: Remember to give controls and objects meaningful names. While not an overly big deal for our little example, when creating more complex games, it quickly becomes a necessity in order to keep track of what you have done.

2.Next let's parent the camera to the car so that we can see where we are going. (Select the camera 1st, then the car and hit "Control + P")
3.We are going to use logic bricks to set up basic driving controls for the car. We will only be creating 4 controls, tied to the arrow keys on your keyboard for navigation. That will give us forward, left and right turns and the ability to go in reverse.
4.Select your car; go to the game logic window (F4).

5.Toggle the “Actor” button, a whole bunch of new buttons appear
6.Toggle Dynamic
7.You can leave the rest of the settings at default for now
8.Toggle “Bounds” and choose “Box” as bounding type
9.Once you have set the bounding type, go back up to the Actor buttons and look for the one that says "Radius", set it to fit around your car model. You don't want it too big or too small. Switch to solid or wire view to see the size of the radius in relation to your model.

Now I am going to show you how to set up the 1st control, then you can set up the rest by looking at fig 16 or checking out the included blend file.
1.You will see 3 sets of boxes to the right of the screen (sensors, controllers, and actuators)
2.Toggle the “Add” box next to the 1st one (sensor)
3.Click on the box that says “Always” scroll down to “Keyboard”
4.Click on the box next to “Key”, it will then say “Press a Key”; press the up arrow on your keyboard
5.Toggle the “Add” box next to the 2nd one (Controller)
6.Toggle the “Add” box next to the 3rd one (Actuator)
7.In the middle column of “dLoc” type (0.25)

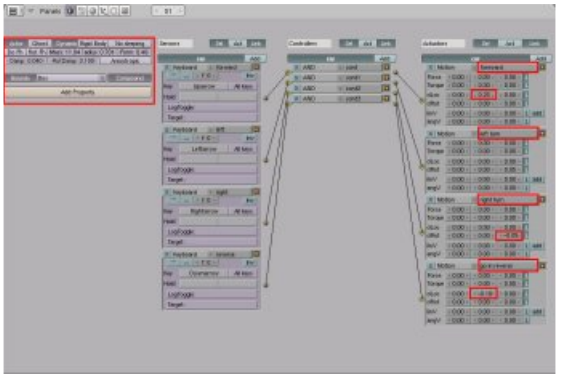


Fig 16: Blender Game settings.

8.Click on the little ball next to the Sensor column and drag it to the Controller column and then from the Controller column to the Actuator column.

You have now connected logic bricks to create your first control. In effect you have told the game engine that you want the car to go forward (or move along the Y axis) anytime you press the up arrow key. That’s pretty much all there is to it. Press “P” to play your gamelet.

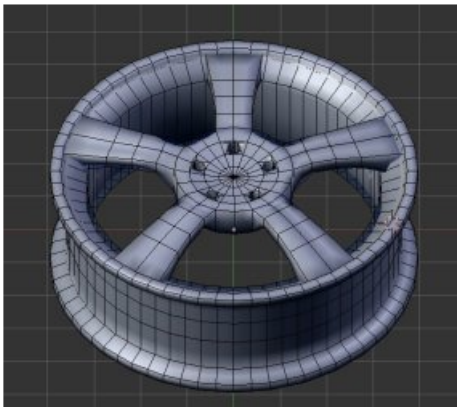
Here are a few suggestions to improve your game;
1. Add sound
2. Engine revving when you push the up arrow.
3. Tires turning when going left or right
4. Thuds as you hit the ground
5. Add moving objects (i.e. cows, pigs, tumbleweeds)
6. Add ramps or obstacles with or without water traps
7. Add falling objects to avoid
8. Add buildings/covered bridges
9. Add lakes in some of the depressions/valleys
10. Drop a few low poly trees in
11. Surround the track/grid with water (turning it into an island)
12. Add a sky

You can find great tips on improving the game environment in Issue #7, “Creating a Realistic Environment for BGE” by John Allie (plantperson).

Further game engine resources:
- http://mediawiki.blender.org/index.php/Game_Engine
- http://www.continuousphysics.com/Bullet/
- http://blenderartists.org/forum/forumdisplay.php?f=12

Modeling A Car Rim

- Pablo Delgado



Level: Beginner to Intermediate

Introduction

Haven't you ever looked at your car and had an idea for a set of rims that would turn heads? Maybe you have gone as far as sketching them out on a piece of paper. Well, now you are going to learn how to bring that idea to life! Well almost. In this tutorial I will teach you a simple technique that you can use to model your very own rims.

I have tried my best to provide tool name and key combinations that I have used, still I assume that you are fairly familiar with blender and know your way around.

Lets Get Started

Once you have decided on your design and you know how many spokes your rim is going to have, then we begin laying out the foundation. For this tutorial we are going to model a five (5) spoke rim. So in order for me to be able to later correctly attach my spokes, I have to create a base whose vertices align perfectly with those of the spokes.

So to do that I find a low number that can be equally divided by the number of spokes that I am going to design, which in this case is 5, but high enough to allow us to give the rim nice details and places a vertex dead center of my spoke. So I chose 25. Create a circle by clicking the right mouse button and selecting "add>> mesh>> circle" and rotate the circle so that one of the vertexes is aligned with the y-axis. Your work should resemble Fig 1.

From the side view select the top vertices using the box selection tool. Now press "E" to extrude, but do not move the extrusion. Simply hit

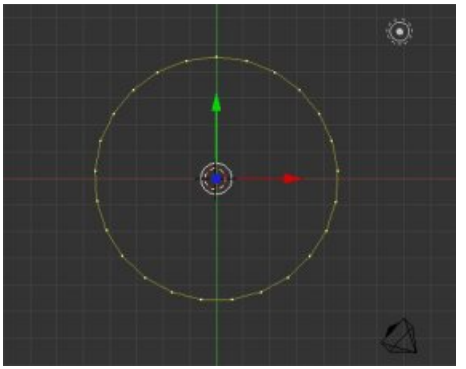


Fig 1: Inserting for the rim.

enter to leave it in place. Then from the top view, press S to scale down the extruded vertices until you have an image that resembles image number 2.

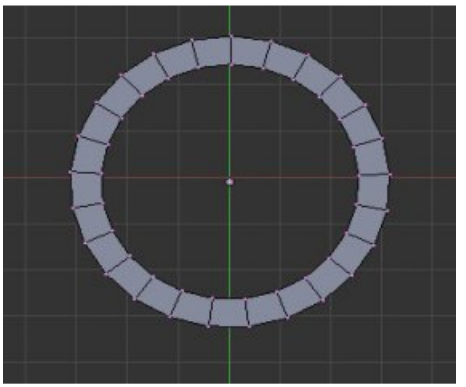


Fig 2: Extruded circle.

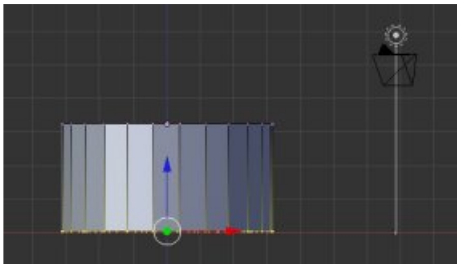


Fig 3: Extruding the rim's height.

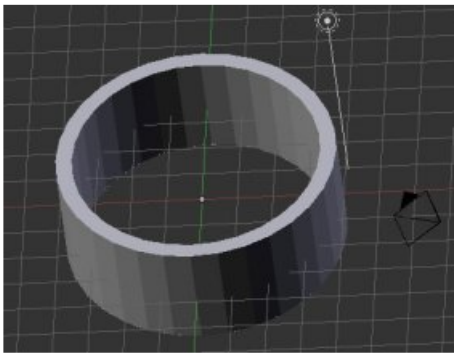


Fig 4: Solid view, rim base.

Select the entire object and press E to extrude again and hit enter without moving your selection. From the side-view move the extruded vertices down a bit until the tube you are creating is the size of a rim. See Fig 3 and 4.

Select the entire object and click the subdivide button from the Mesh Tool. This is so we have more polygons to work with when we begin to develop our spokes and lip. From the side view select the line that runs around the tube and move it up close to the top. See Fig 5.

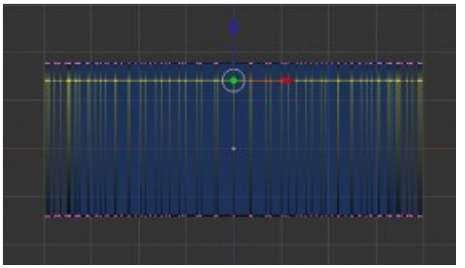


Fig 5: Selecting tube line.

Select the entire edge of the inner tube at the top by selecting one edge and pressing “Crtl+E” and selecting “Edge Loop Select” from the menu. Now from the side view bring it down close to the edge that we earlier moved up near the top. Press S to scale the edge a bit so that it looks like Fig 6.

Now we are going to model the walls around the rim where the tire goes. Select all the faces along the outside of the tube. Make sure you do not select the faces from the inside or you will be in for some headaches. Use the knife tool (Shift + K), select multi-cut and enter the value 3. Your work should resemble Fig 7.

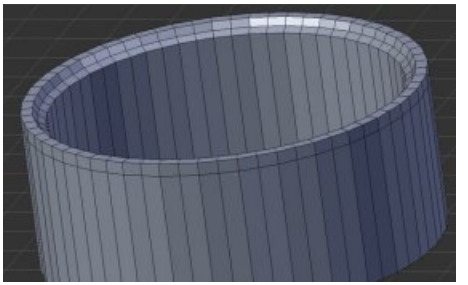


Fig 6: Solid view.

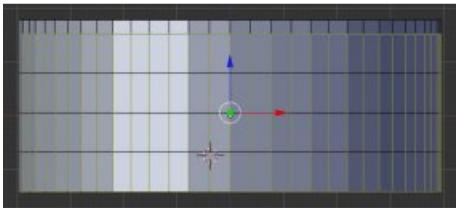


Fig 7: View of rim after inserting mesh cuts.

Select the last edge of the tire, near the bottom of the tube, and move it close to the bottom about the same as the edge we moved toward the top. Move the other two edges we created and center them so that we have two large rows running around the rim, then select all the faces. See Fig 8.

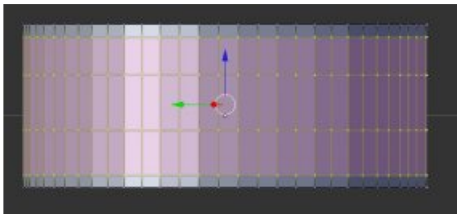


Fig 8: Edges loops for the rim.

Scale down those faces so that it creates a lip near the top and bottom of the rim. See Fig 9.

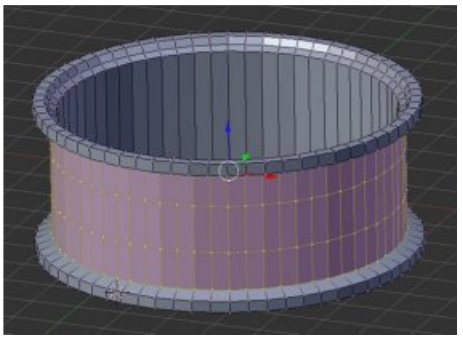


Fig 9: Rim placement, user-view.

There is one final detail that we need to do before we begin work on the spokes of the rim. Select all the faces of the inside of the rim that falls below the edge we had moved up. Press S and scale them down just a bit so that they make the inside walls smaller. Now use the Knife tool to make a new edge and move it down so that it resembles

Fig 10. Finally, the base of the wheel is complete! Now we can start working on the spokes.

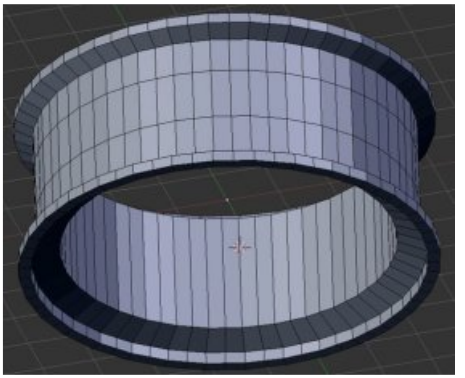


Fig 10: Rim base pulled down.

Center your cursor and add a plane. Move the plain so that the center of the planeis in the bottom left corner. On the modifiers panel add a mirror and enable clipping. Pull the top of the plane until it enters the walls of the rim. Then select the plane and move it to another layer. See Fig 11.

Select Knife >> Multi-cut and enter 5 and cut the plane horizontally. Move the points to give the shape of the spoke. See Fig 12.

From the side view select the plane and extrude it upwards to give the spoke its thickness and apply the

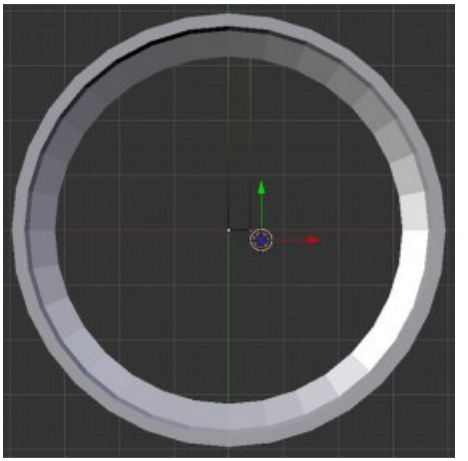


Fig 11: Added a plane for spoke.

mirror so that it becomes permanent. See Fig 13.

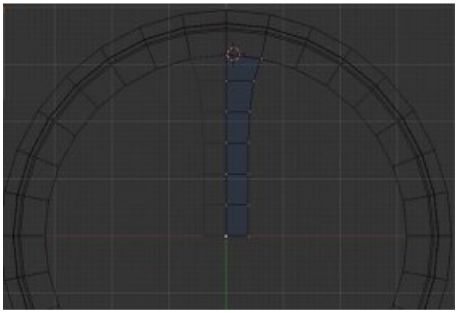


Fig 12: Spoke taking shape.

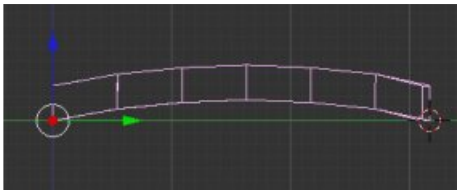


Fig 13: Curved spoke.

Hurray, we have our first spoke - four more to go. We can do this the hard way (By eye balling it) or we can do this like pros and do a little math. A circle has 360 degrees and we divide that by 5(the number of spokes we have), so we get 72. Which means we are going to lay each spoke at a 72 degrees angle from each other. So press tab to exit edit mode and enter object mode. Press “Shift + D” and press enter. It seems like nothing happened but something did, it created a clone. Now take your hand off the mouse and press “R”, to rotate, and using your number pad press 7, 2 and hit enter. If you noticed the clone is now visible to you and is laid out 72 degrees from the first one. Repeat this step three more times so that you have five spokes. See Fig 14.

Lets go back to the first layer and start creating the center of the rim. Create a cylinder and scale it down so that it looks like Fig 15.

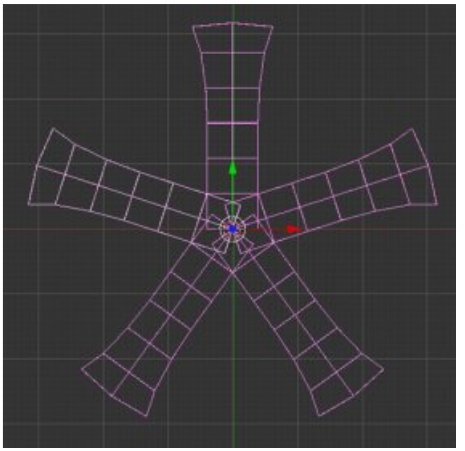


Fig 14: Spokes.

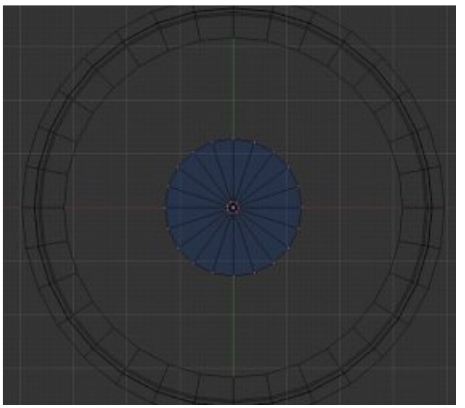


Fig 15: Added a cylinder.

Select the vertex in the center both from the top of the cylinder and the bottom. On the proportional drop down menu, which is the one that looks like a donut, select connected and on the fall off menu, which is the drop down right next to it, select “Sphere Falloff.” From the side view use the arrow manipulator to pull the two vertices down until you create a concave shape. See Fig 16.

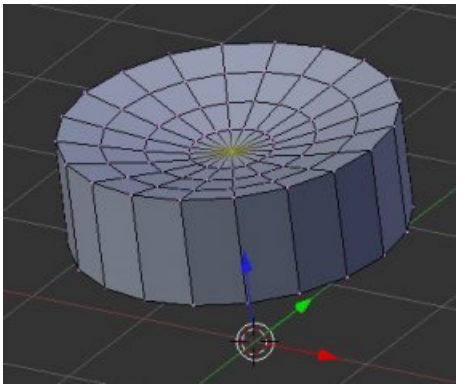


Fig 16: Concave shape.

Go back to the second layer, where you have the primitives for the spokes. Select the bottom four cubes on each spoke and delete the faces. Move the center of the spokes from the first layer to the second, where your spokes reside. See Fig 17.

This next part is a bit tedious but just as fun. Select both the cylinder (Rim Center) and the spokes and press and “CTRL + J” to merge both objects together. Now begin to merge each vertex on the spokes with the corresponding vertex on the cylinder so that the resulting image looks like Fig 18.

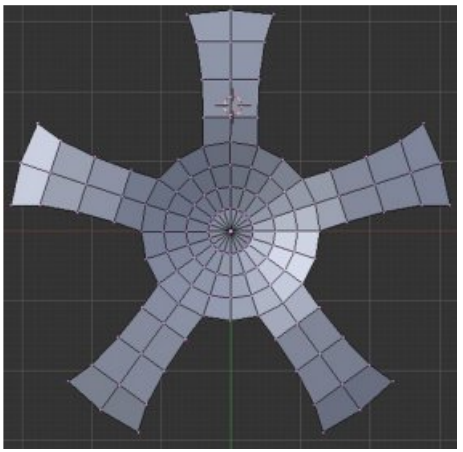


Fig 18: Spokes on the centeral rim.

Now that we have the shape of the spokes, we have to create the holes where the bolts go. Create another cylinder and scale it down so that it looks like a long rod with the thickness of the holes we want to create. Place it near the top spoke and center it on the y-axis and so it traverses the spokes center. Place your 3D-Cursor at the center of the screen (x=0, y=0, z=0) and change the pivot point to 3D cursor. Press tab to exit edit mode and select the rod we just created and duplicate it by pressing “CTRL + D” and press enter. Press “R” and enter 72 on your number keypad. This should have moved the clone rod to the exact location as the first but on the second spoke. Do this three more time until you have 5 rods. (Note: remember to change the pivot point back to its normal state or you will get frustrated later.) See Fig 19.

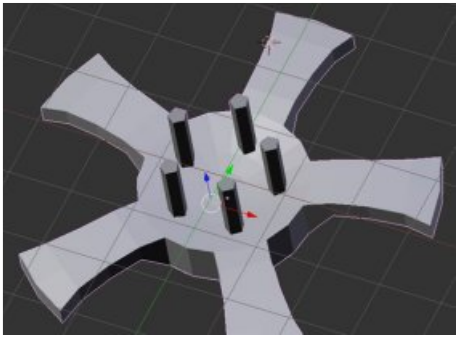


Fig 19: Rim bolts.

Next Press “W”, which brings up the Boolean Tools Menu, and select “difference” from the menu. This will create an object that is composed of all the meshes that do not intersect. Press “G” to grab your newly created object and move it off to the side. Select the previous two objects that you had and delete them, and center your new spokes with holes.

We are almost complete, so hang in there. Move the newly created object to the first layer. Move the spokes toward the top of the rim. Now that they are in place select both the rim base and the spokes and press “CTRL + J” to join the to meshes into one object. Begin to merge each vertex on the spokes with the corresponding vertex on the rim base near the bottom of the lip. See Fig 20.

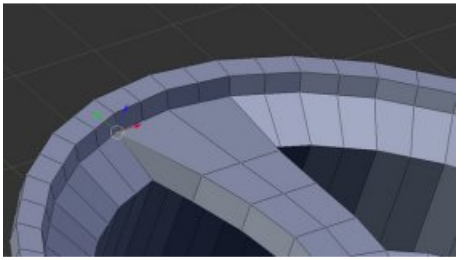


Fig 20: Joining spokes to the main rim.

All that is left is to add detail so I add a subsurface modifier and use “SHIFT + E” to make certain edges around the lip of the rim and the edges of the spokes are sharper. See Fig 21.

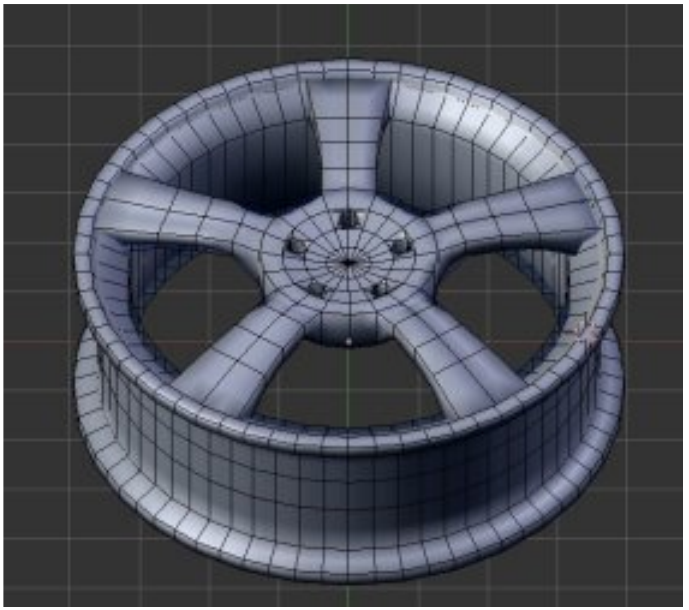


Fig 21: Spokes on the centeral rim.

Rendering of the rim we just finished.

Texture, Position and Render to your hearts content !!! Enjoy !!!

Pablo Delgado

Pablo is 25 years old from Puerto Rico currently living in Miami, FL. He works as a software engineer and does part time 3D work for customers using Blender and Lightwave.
www.pdelgado.com
pablodelgado@bellsouth.net

Modeling Tires

- Thomas Baron

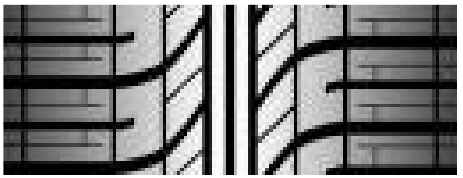
Level: Intermediate

Introduction

In this tutorial I will take you through the steps to model the tires like the one I had earlier made for an Alfa Romeo GT model.

What you will need

A sketch of the tire, like this one (PIRELLI P6000). For the dimensions of the tire, let's use a 205×60r16.



Overview

We will buld the dimensions of the tire from the image we have so lets calculate that:

The width of the image = 205mm.
The external diameter of the tire
16inch + 2 * 60mm =
16*25.41+2*60 = 526.56mm
From the image size (in pixels: 500*190), We calculate the height of the image (a part of the tire tread) to 205*190/500 = 77.9 mm

As the diameter of the tire is 526.65mm, its perimeter is 3.14159265 * 526.65 = 1654.52.

So, on our tire tread, we can put N times the sketch image with N = 1654.52/77.9 = 21.239.

Because the image can be tiled, we need to round off the fractional number (21.239) to a whole number. So we calculate out that the tire tread sketch we have will fit 21 times around the tire.

Create a 21-edge circle and extrude it to have one face fitting well over the sketch. Next, we'll model just one element (1/21 of the full tire). and then multiply it 21 times, merge it, and voila!

Step1. Scene setup

Open Blender. Erase all elements of the default scene.

In the Right view (NUMPAD 3), add a Circle mesh of 21 vertices. Rotate it to have one of the bottom edges perfectly horizontal (tip: align top, opposite vertex with the Z-axis.

As the Blender default Circle's radius is 1.414 (square root of 2), size it by 0.707 (1/1.414) to have a radius of 1.0. Then, size it by 0.526/2 = 0.263 (tire radius in meters) and move all vertices up by 0.263 (the radius).

It's OK that the bottom edge is not on Z=0. We'll see why, later.

Duplicate this bottom edge; separate the duplicated vertices to have another object. Extrude it along the Y-axis by 0.205 (the width of our tire). You now have a Plane and a Circle like these:

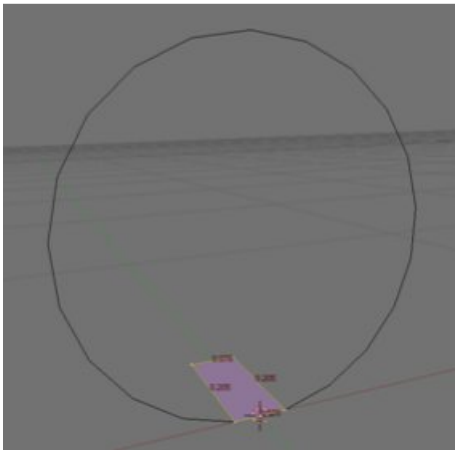


Fig 1: Tread prepration.

You can check that your work is OK by displaying edges length: you should have 0.205*0.078 for the plane. One more step and you're done: duplicate the Plane, change one to display only as Wire (Drawtype in Object panel, F7) and then, select the other one. Switch to UV Face Select mode, go to the UV/Image Editor and load the sketch image. Return to 3D view and Object mode. Activate the Textured draw type (Alt+ZKEY), you get the following:

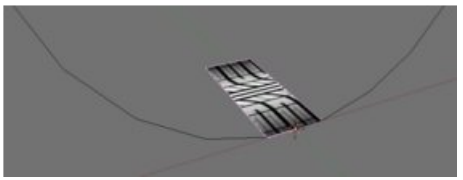


Fig 2: Tread image.

The sketch is displayed correctly, without any distortion: the few math calculations we did before starting Blender were not useless.

Step2. Modeling in parts

Select the Plane that we set to display as Wire. We'll work on it to model a sub-part of the tire thread. First, use symmetry to save yourself some work. The sketch we have can be split along both the X-axis (radial cut) and the Y-axis (axial cut). So, you can subdivide the Plane one time. But before you do, carefully read the following.

When splitting your edge along the Y-axis, no problem. But when splitting along the X-axis, remember you're not working on a Plane but a section of a cylinder.

That's why the bottom edge of the circle is not at Z=0. Let's experiment to get a clue. If you've already subdivided the plane, please undo it. Then cut the face (CTRL+ RKEY) at percentage 1.00. Don't select/deselect anything to keep the new vertices selected.

Then, rotate carefully these vertices around the Y-axis (RKEY then YKEY) to get them at X=0 (as seen from the Top or Right view) And in the Right view (NUMPAD 3), you can now check that these vertices are almost at Z=0.

Remember this move: when cutting along the X-axis, do not move edge vertices along the X-axis, but rotate them around the Y-axis, to move them around a perfect circle.

Just a little note: you have to move the Plane, textured with the sketch, downward to get the added vertices above it (so they're still visible).

Now you can continue with modeling. Remove 3/4 of the plane as said above, and cut the plane according to each detail visible on the thread. You should get something like the image below.

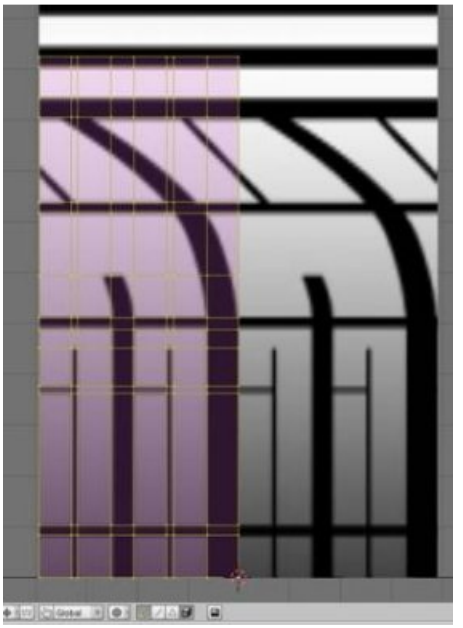


Fig 3: Tracing the treads in mesh.

As you can see, cuts in the plane are mapped on thread details, but not on the curved details.

To solve it, just rotate them around the Y-axis. This time round details are correct.

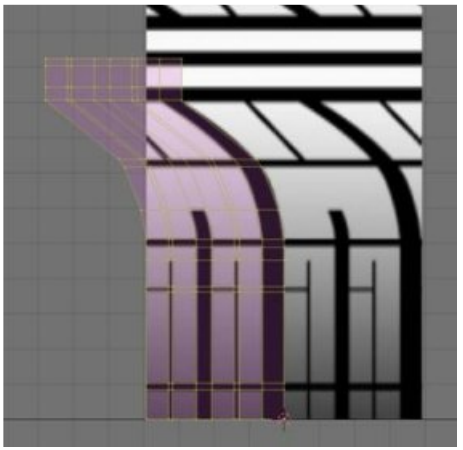


Fig 4: Rotated placement of elements.

As you can see in the Right view, the tire element is not a Plane but warped around a circle.

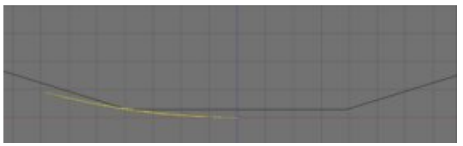


Fig 5: Adjusting the tread curve.

This is why it's so important to rotate vertices around the Y-axis instead of translating them along the X-axis. Now, we will cut the edges in dark areas, to materialize the hollows of the tread. You should obtain this:

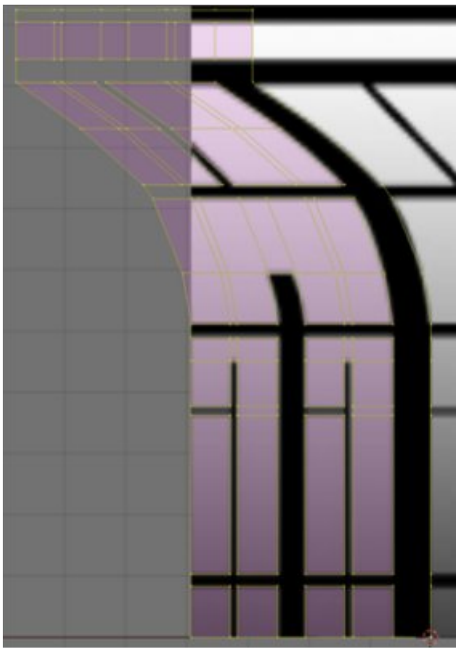


Fig 6: Corrected placement of elements.

It's important to keep the edges circling the dark areas! We will see why later. Now it's time to optimize our mesh a bit. We have to remove all unneeded vertices and edges in white zones and on borders. Doing this will allow us to save a lot of polys on the completed tire (remember, we will duplicate this part 2*2*21 = 84 times !!!).

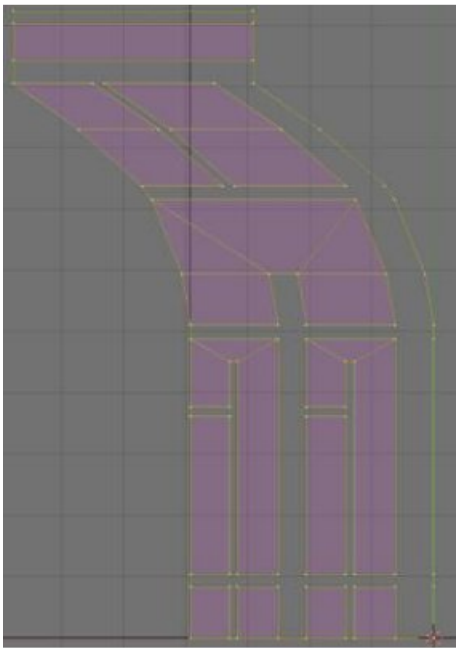


Fig 6: Optimised tread element.

The vertex count goes from 144 to 78 and, the face count goes from 53 to 22.

Step3. Adding some thickness.

Time to model the hollows! From the previous step, select all vertices

Step4. Completing the tire

SpinDup the element you patiently modeled (1 turn, 360? of course, 42 steps: because we need 21 elements and the mesh we have is only a half). Once SpinDup is done, you may observe your border vertices are not perfectly duplicated. So, select all vertices, and Remove Doubles (set threshold to 0.001).

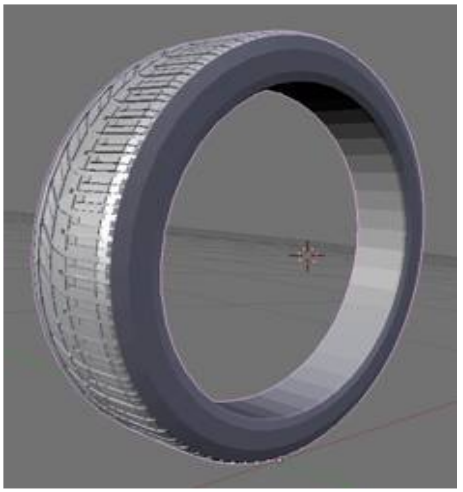


Fig 11: Half of the tire.

You now need to mirror the tire. Select the row of vertices at the middle of the tire, and move the 3D Cursor to selection (to have it at the tire's center, in the middle plane of it) Duplicate the half tire, don't move it, but rotate it 180 degrees in the Top view (around the 3D Cursor). Rotate around the Y-axis to align the middle vertices to the ones of the other half tire. Merge, and you're done!

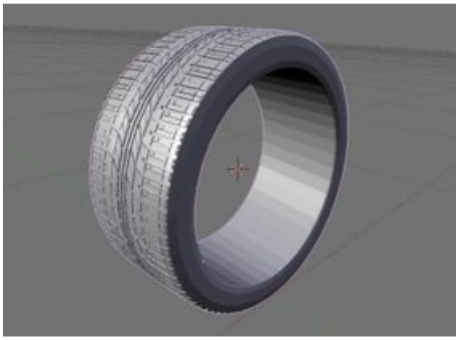
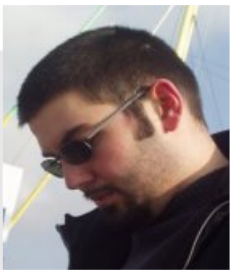


Fig 12: Completed tire mesh.

I advise you to use this mesh with AutoSmooth on, to have solid thread details as well a smooth side. Here are some Yafray renders with a rim (black tire material courtesy of Sonix's Car Material Library).

Thomas Baron

Thomas Baron is a software engineer for the aeronautical industry in France. After discovering Blender in 1999 he has been a serious amateur user for two years. A car enthusiast who also likes to work on planes.



Making Of Scale Model

- Marin Myftiu

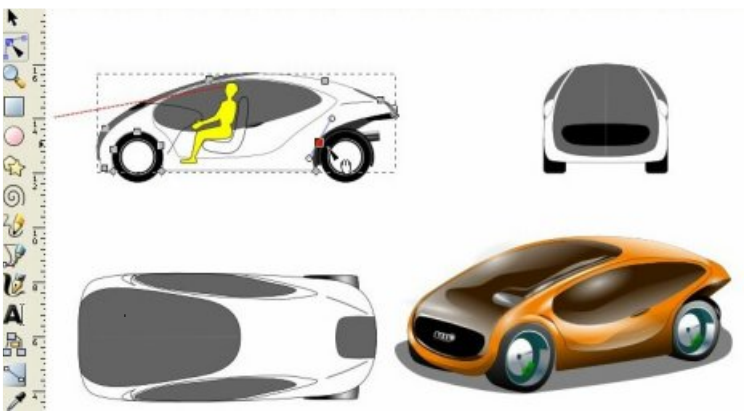
Introduction

"Sharing the Road," is the theme for the 2007 Michelin Challenge Design™ (MCD) (www.michelinchallengedesign.com), a year-long transportation design event that culminates at the North American International Auto Show (January 13-21, 2007) at Cobo Center in downtown Detroit. This year's entrants - over 260 of them from around the world - addressed cars and other light vehicles, semi trucks and passenger buses as well as road users on two wheels and two feet.

Bob Miron, Michelin's director of technical marketing, notes some of the design solutions selected by this year's eight-member jury - an expert panel of renowned designers, transportation industry educators and safety experts - addressed being "doored."

"The xV concept from Marin Myftiu of Albania features a locking system that prevents a door from being opened for a few seconds after the vehicle is turned off when an object approaches within the door's span". This tutorial covers how Marin Myftiu modeled/created his entry.

Car design technique;
Concept to scale model.
Getting started.



During this tutorial, I will explain the whole process I have followed for designing a car. It will not be a detailed, step by step method since it covers many fields beyond Blender or even your own PC. This is an overview of the whole process, with occasional focus on particular techniques. We will start with the concept sketches and end up with a 1:6 foam scale model.

Concept: First of all it takes a lot of thinking -> sketching ->rethinking -> re-sketching. Car modeling is the most frustrating thing I have seen in 3D modeling up to now and you've got to have crystal clear ideas (or almost) of the ins and outs, solid and voids, bumps and depressions of that surface before you start off; most frequently, big mistakes found during 3D modeling will require you to redo most of the work, that may be whole weeks. So first of all, do a lot of sketches and dif-

ferent views to clarify as many details as possible.

Blueprints

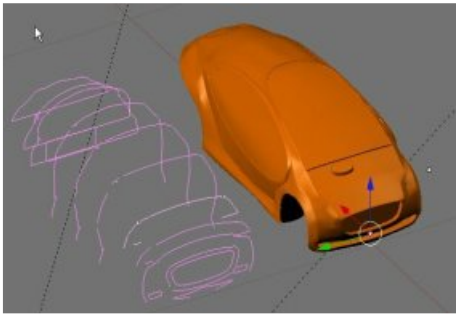
As you might have seen in car modeling tutorials, it always takes blueprints of that particular car and the same is needed for your own concept. For some precise blueprints you can use freeware vector software like Inkscape (like I did). Inkscape also helped me go one step ahead of my sketches, making a colored illustration. When working

on the blueprints, I would advise to keep an existing car blueprint on the background to keep the proportions in check.

Blendering

Well, assuming that we have done all the above correctly, it's time to warm up the Blender engines. I am on a 64bit AMD machine and 64 bit system so the x64 / SSE3 optimized build gives a lot of added rendering power (+75%) compared to the standard. I will not cover here, in detail, the whole modeling as it is a tutorial apart but, a few concepts are almost vital when turning your concept into a 3D:

- *Part Subdivision.* Always subdivide your model body into distinct parts (doors, sides, cowling, mirrors etc). Attempt-



ing to separate some parts after having done some modeling, by cutting or other methods, will cost you a lot of time and efforts to correct/adjust them.

- *Grid Resolution.* The same is to be said when you model the polygons. The technique I have used is pretty straightforward; Extrude a point following distinct curves on the background blueprint. Neighboring curves should have approximately the same number of vertices to form as many regular, four vertex faces as possible. You have to set in mind the exact positioning of the main curves since any later *cut* or *loop cut* may seriously damage the surface.

- *Correct Lighting.* Usually I use a traditional 3-lights scheme and play with them to get good lighting and since most often time is a concern so I don't include radiosity or occlusion in test renderings.

After all the work, the final renderings looked something like this:

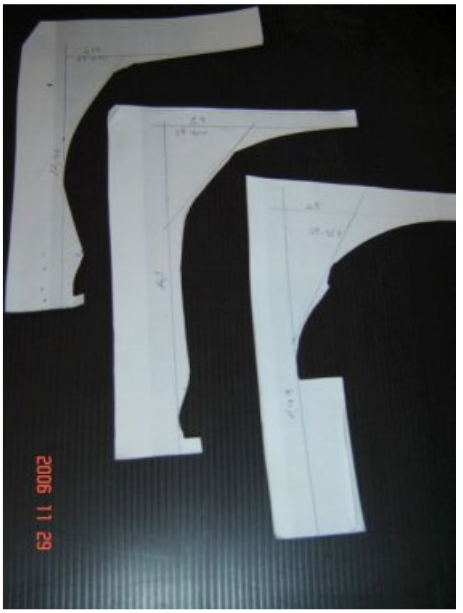
Scale model

When finished with the renderings

and photo editing, it was the time to start thinking about the scale model. While I was not an expert with Blender, a scale model was something in which I had zero experience and all I had at my disposal were simple tools like a saw, shaping gauzes, scissors, rulers, a simple caliper and few other things.

The first thing I did was to print all the three main views of the model in 1:6 scale (the model's real scale). These images on the wall served throughout the modeling process as a 1:1 reference for most of the dimensions; you open the caliper according to the part you want to measure and then compare it with the real dimension on the model, pretty straightforward.

Another trick in which Blender helped me, was preparing cardboard templates of some given body sections. I subdivided the whole length of the xV in 12 sections and cut the car body after those sections. After rendering each section cutout, scaling them in the GIMP and then printing, we only have to stick each section on a cardboard piece and carefully cut the hollow part.



After this operation, the only thing to do remains marking the sections on the working board according to the distances of the sections we have in Blender, so we can place the right section at the right distance to check the work progress.

And this is the final output. This model will be displayed by Michelin on the North American International Auto Show, in early January 2007. As you could see most of the process, Blender played a crucial role not only in modeling and rendering but also in putting together a correct scale model and the result was pretty satisfactory.

Marin Myftiu

Marin Myftiu is from Tirana, Albania. He is currently studying Architecture and working part time as a graphics designer. He is working in architecture, product design, interior design projects and more importantly seeking a career in car design.
www.freewebs.com/multid

Making Of 'Cutting The Waves'

- Zsolt Stefan

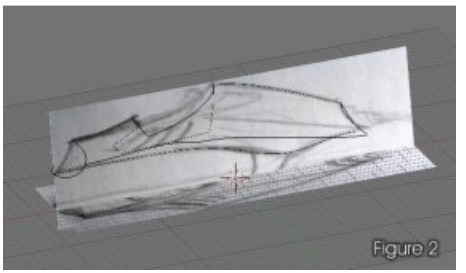
Making of 'Cutting the Waves'

As a regular Blender user, I have always waited for the annual Blender F1 challenge. Unfortunately for my 2005 entry I had little time and couldn't complete it. Basically, the car was done but I had no time for a background scene. Still, a fifth place motivated me to try harder next year, so for 2006 I decided to make the best image I could, with as much detail as possible, including a background.

Modeling the car

This year I wanted something fresh and new. So I ditched the old wheel-based car concept and decided on a hydrofoil-type vehicle with a jet engine. This combined flight and water, as I really wanted something based around the theme of "water" this year. I started sketching and decided on the following design (Figure1):

As with most projects, this one also started with the modeling phase. I scanned my drawings and loaded them onto two planes in Blender with UV mapping, giving a top and side view, as blueprints.



I usually model with subdivided meshes, as the mesh tools in Blender are the most advanced. However, to aid in modeling, I first drew the outline of the car with 3D Bezier curves, adjusted to fit the top and side blueprints (Figure 2). Then came some adjustments in 3D to make them nice and smooth, giving the vehicle the proper aerodynamic outlines.

Using the curves as outline guides, I mesh modeled the car body. Then: Subsurf once, Alt-C convert to mesh, and continued modeling. Figure 3 is more or less the final model, with front grille, windshield, roof air intakes and front lamps added. To make it more interesting, there is

an indentation, dividing the fuselage into two differently coloured parts, purely for aesthetic reasons.

The Driver

Though I am not much of a character modeler, especially not with realistic humans, I just had to add a character to the scene to make it come alive. Figure 4 shows my driver, along with the insides of the cockpit. His body is my model, but I have to admit his head is a MakeHuman head stitched to the body. He's fully pose-able, except for fingers. Following my motto of "detail, detail, detail", I added helmet, glasses, 4 point seat belts with the release button modeled, though you can't see it in the final render.

Water

The water I used comes from an earlier picture of mine (Venice Morning), as I liked it so much :) Its really nothing too complicated, the waves are

done with three Clouds textures, with different Input Size X,Y,Z parameters as bump maps, giving 3 layers of waves: small, medium and large. The material itself is a simple greenish blue reflecting and refracting material. Figure 5 is my first WIP posted to some 3D graphics forums on the net (Elysiun/Blenderartists). This is good practice, as the comments I received really helped me develop the picture further. Most people said it looked like a small scale model and not a full size race

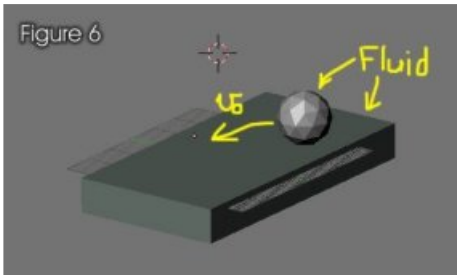
vehicle, like a toy in a bathtub. I decided to get rid of the "bathtub" and go for the open sea!

Lighting

For lighting, I used an HDR map generated with Terragen, as Terragen can make some very nice sky renders. To find the perfect mood for my image, I tried various setups for the HDR maps until I found the central one as shown in the following picture, a very nice, pinkish-yellowish morning sky. It was undoubtedly the 'One' perfect for a morning test run on the open-sea track! Besides the global GI lighting, as with many of my images a sun lamp is also used to add some stronger shadows, though these do not show up well on the water. For realism, the sun lamp is aligned with the sun in the Terragen HDR map.

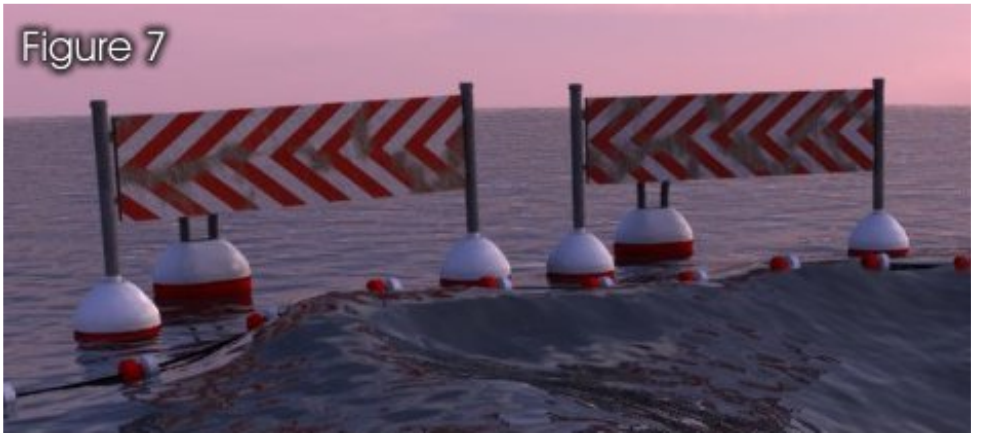
Fluid Waves

One important part of the picture is how the sea reacts to the hydrofoil. The vehicle is in constant motion, **"cutting the waves"**. In this captured moment it is taking a sharp left turn, while the water waves follow the original path of motion. To create the proper swelling and depressing of the water surface, I used the nice Blender feature of fluid sim-



ulation. It took at least 20 or more tries to get the proper waves to form. The setup that finally worked was a cube shaped block of water at the bottom of the simulation domain, and a large ball of water is 'thrown' with a given starting velocity (v0) sideways into the water, in the direction that the vehicle was going. This created a very nice wave effect.

Next step was integrating the bounded fluid domain into the big disk of water that was the sea. After some unsuccessful tries with other tools, the built in script: Mesh -> Apply Deformations converted my baked fluid-sim into an editable mesh. This was stitched to the large round sea mesh. TIP: to make it seem like it extends forever (till the horizon), the outside of this disk is about 1000 Blender units away. This is not enough, you can still clearly see that the 'world' ends abruptly. So the outer vertices were moved up a bit, making the disk curve upward a bit, and it therefore seems to extend to the horizon.



Extras

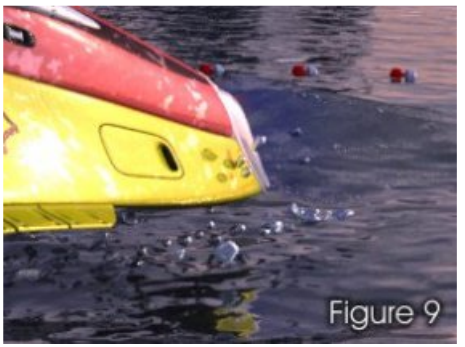
With the main objects done, time for the extras! I modeled two dolphins and painted them with vertex paint. You can spot them in the water in the final image. I'm not exactly sure a race track would be allowed where dolphins lived but I just had to keep them in the final image. The race track itself is bounded with buoys and some direction boards (Fig. 7). There's also a camera on a platform (Fig. 8), the same man model from before. It was fun to model his movable pedestal, which is properly rigged with an armature, with the wires and everything following its movement.

Figure 9 is a closeup of the vehicle's back, with several interesting details. There is the jet engine exhaust which properly 'warps' the air - the refractive index of hot air is higher than of cold air - , and has just a hint of blue where it leaves the car. A common mistake that I see in many pictures is excessive use of thick black or fire coloured fumes, usually particle systems, for jet engine exhausts. This is unrealistic, engines don't burn air, if you see fire and/or smoke - not to be confused with condensation trails - coming from an exhaust, there is serious trouble in the engine (reminds me of: Monty Python, How to Irritate

People: "There is no cause for alarm. The wings are not on fire." :)), so I'd advise against it. The see-through exhaust here was actually a pretty complex material, with a number of procedural textures and colorbands used. The small water drops are the result of a separate fluid-sim, that was converted to mesh with the aforementioned script, and just the small drops kept. There are also bright spots on the hull of the 'car', these are meant to be water drops that have stuck to the fuselage at the back (the air resistance would blow the drops off the front).

Final touch-ups

When the date of the competition was extended 1 month, I didn't really have anything left to do. I added the island with the lighthouse in the background, with some palm trees, the details of which were unfortunately lost with the motion blur. Right: motion blur! Basically I rendered a depth map (ie z-buffer) of the image, and using that as a mask, applied post-pro motion blur in Corel PhotoPaint to the back parts of the image. This was a simple way to separate the car from the foreground. I also applied some Gaussian blur and desaturation to the background.

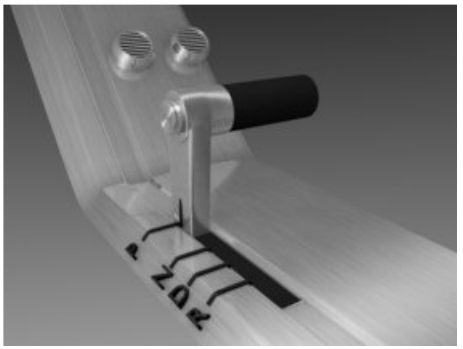


Trivia: why does the side say "Falcon Racing"? If you look at my blueprint sketch at the beginning, the car looks like the open beak of some bird of prey. Hence "falcon" and the logo as well.

And the final image: Placed fourth. Not bad, but could have been better. But there's always a next time! :) See you at the races!

Textured Metal Shaders For Industrial Design

- Claas Eicke Kuhnen



Level: Intermediate to Advanced

UV-unwrapping - material nesting - displacement mapping - displacement mixing

Introduction

This tutorial will explain how different types of metal surfaces, found in Industrial Design and Metal Art, can be rebuilt in Blender. It will cover brushed and hammered metal textures including modeling, UV-unwrapping, and texturing approaches and techniques.

Most of the scenes are jewelry related because this is my main field at the moment. However, everything explained can easily be transferred to any other type of object as well - as the last chapter shows.

1 Brushed metal shader

Let's talk about a brushed metal shader first.
Many first semester students marvel about their first casted silver ring after they've started to clean up their first casting with sand paper. They love to move the ring around and enjoy watching the change of those stretched highlights. But what are those anisotropic reflections? Micro fine grooves inside a polished metal surface reflect the incoming light at different angles away - perpendicular to the grooves direction.

Those visual elements, most commonly known as linear and circular brush patterns, are called anisotropic specular reflections, or short anisotropic highlights. However, those grooves also spread the reflection value in the same manner. This is known as anisotropic reflection. It is not just a diffused, radial blurred reflection, because the anisotropic reflection is stretched as the specular highlight reflection is stretched. An anisotropic reflection is composed out of many fine parallel reflection lines, depending on the groove size.

It is important to know that close objects appear more clear in the reflection and far objects appear more diffused. The last point is especially what is currently difficult to recreate with the Blender material system. It is not only a question of stretching a reflection along one axis perpendicular to micro-fine grooves, but furthermore to make that reflection have the correct blurriness according to the distance between the reflected and the reflecting surface.

To create this visual effect we need to create a bump map which can provide us with the needed amount of fine grooves. Because we work in 72 DPI and not the real world, we only have a limited amount of visual information (pixels) to create this effect. The solution to this technical problem is to create a blend of different materials which all have slightly different bump values. By overlaying and mixing them together we can create that fine surface we need. Each material we are going to create has a small reflective raytrace value together with a desired Fresnel setting. In addition,

each material is utilizing the UV mapping mode for any used bump map.

The first material contains the main anisotropic specular reflection utilizing the new Tangent V function for specular highlights. This functions as the main base material. It also contains a low setting for raytraced

Fig 1: Base tangent.

reflections. It is very helpful to give your materials unique names to help you manage them.

In terms of texture you can either use an image map you create inside Photoshop (or other image editing software) utilizing motion-blurred noise, or you can use a procedural function. The benefit of a procedural texture is that it is resolution independent. I prefer to use voronoi. This texture can produce nice point patterns similar to noise. Using the the Int function you can decide if you want to use only dots or more a cell-like structure by playing with the Brightness and Contrast value.

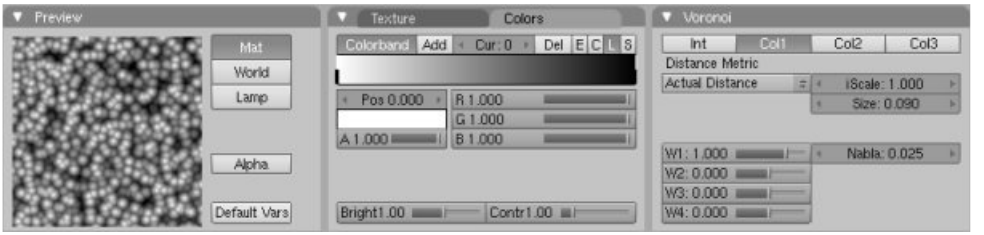


Fig 2: Voronoi Pattern.

In addition, you could experiment with the Col1 option. With the internal texture scale value you can control the size of the points which will later be stretched.

You can now decide if you want to stretch the texture completely along one axis by turning off one axis inside the Map Input tap, or you can use the X-Z size values to scale the

Fig 3: texture Mapping.

pattern down along one axis and stretch it along the other axis.

This will produce the same visual result you would get with Photoshop anyway. The material is connected to one color input channel of a color mixer node which blends into the color channel of the output node. The next step is to create a second material that contains a diffused unstretched specular reflection. This helps to build a multifaceted overall specular reflection which is typical for brushed metal surfaces.

This material we duplicate with each copy having a slightly different bump setting, then mix them together with color mixers. The end-result of this material network needs to be connected to the second color channel input to which the first material is connected. The amount of second materials will define the smoothness of the brushed metal look and also the smoothness of the anisotropical reflection.

A schematic view would look like in the Image below:

Brushed metal shader tree: base

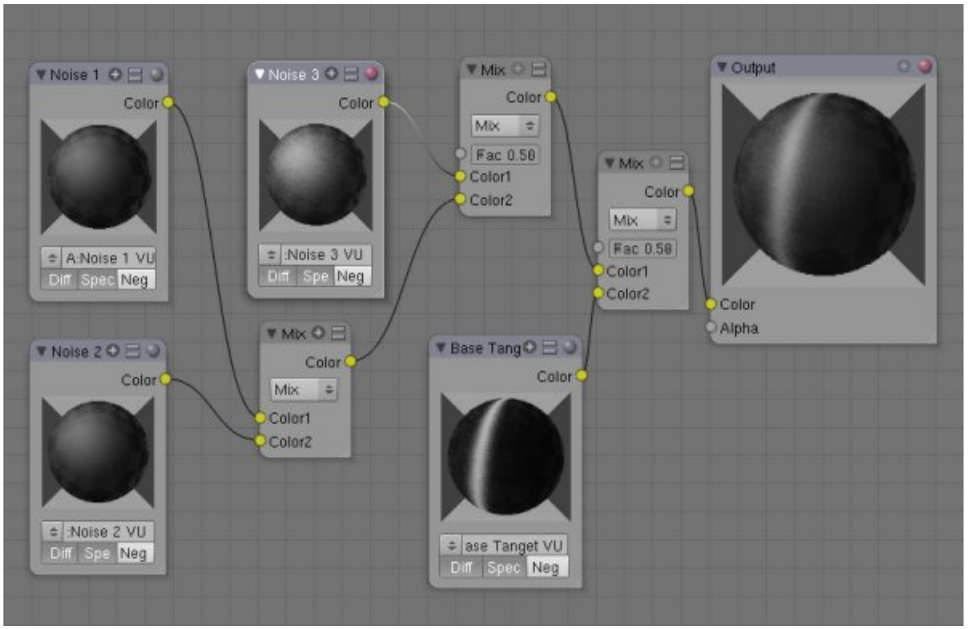


Fig 4: Material Tree.

shader with anisotropic specular second shader with sharp specular and amount of secondary shades with different bump settings

Sealed surfaces

Brushed metal surfaces sometimes have the problem of catching dirt because of the fine grooves. To prevent this you can seal the surface with a lacquer. This will result in a very strong and sharp specular highlight and a more soft looking anisotropic specular highlight. To get this result you need to lower the value of the anisotropic highlight and increase the value of only one regular specular highlight.

How to unwrap

Anisotropic highlights are often found on metal objects which represent geometrical primitives like a tube, a ball, a sheet or a box. Before we continue, I would like to present few important tips:

1. Always try to unwrap the mesh from the side, or top view, but not from any perspective view to prevent any distortion of the UV texture you will apply. If your object or faces are not aligned to any axis, try to look perpendicular onto the faces you unwrap.
2. To prevent distortions, always try to scale faces inside the UV editor to the same dimension they also have inside your 3d model. This is very important to keep in mind when you follow the next steps.
3. You should hide the UV seam by either putting additional geometry over it or by rotating it away from the camera.
4. Use seams to control where the mesh unfolds to have better control about where you will see the highlight break.
5. Rotating a mesh in the UV editor will control orientation of specular reflections. Think about UV in only one direction. 'UV' could be upwards/downwards, and a 90 degree rotated version, lets say 'VU', could go left and right.
6. To create smooth and even-flowing anisotropic highlights on curved objects, you will need to

have a very smooth surface using either smoothed subdivisions or the subsurface modifier. Otherwise you will sometimes see sharp corners in the highlight.

Plane: Unwrap > Project from view: This is the easiest object to unwrap. Just make sure you look perpendicular onto the plane.

Cylinder: Unwrap > Cylinder from view: To texture a closed cylinder, I found it easier to create a tube and unwrap it first. After that step you should create a new loop cut very close to the edge of the end you want to close. Because you unwrapped the tube, the change in the mesh will already be updated in the UV unwrapper's mesh. Select the top loop of the tube and scale it down to close the tube by creating a cap. Do not merge the points. This way you can create a cylinder where the sides have a linear highlight pattern and the cap shows a circular high light pattern, similar to what we know from a CD.

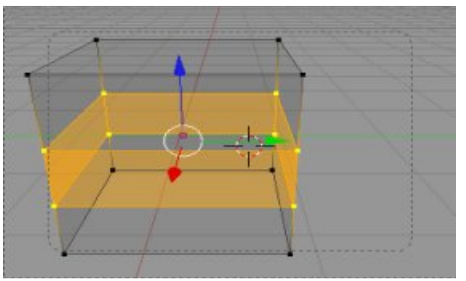
Box: Unwrap > Unwrap with seams marked before: A box is a pretty simple shape to unwrap. You only have four sides and two caps. However in contrast to a cylinder, the box can only provide you with linear highlight patterns. This also means that the cap will show a linear direction which flows from one neighbor face through the cap face to the other neighbor face. The other two faces will provide you with an anisotropic specular highlights which will not create a joint with the specular highlight of at least one cap face. If you take a small steel cube and try to brush each side with sand paper you will see that you can only connect 4 sides like a face loop. The other two faces will not be able to be integrated.

Sphere: Unwrap > Sphere from view: A sphere is very easy to unwrap as well. Just go to the side view and unwrap it. You will notice that the poles show some darker artifacts. To fix that, remove the two top pole points to further open the sphere. Select a loop ring, extrude, and scale the selection down to close one pole. You should move the newly created loop ring to the same position of the vertices you deleted first to retain the sphere curvature at the poles. As with the cylinder, do not merge the vertices ring to close the sphere.

Mixed shapes: Depending on your object you will have to utilize different unwrapping methods to get good results. In my test object I want to unwrap a cylindrical object which contains an extruded element.

Step 1: Building basic body:

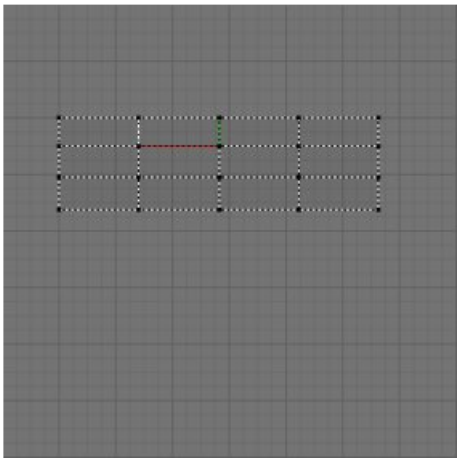
Because you need to have a very smooth surface to receive smooth anisotropic highlights, I prefer to create cylindrical objects out of cubes utilizing the subsurface modifier to create the roundness. Starting with a box, I remove the pole caps and include two loop cuts. (Image 1)



Img 1: Loop cuts.

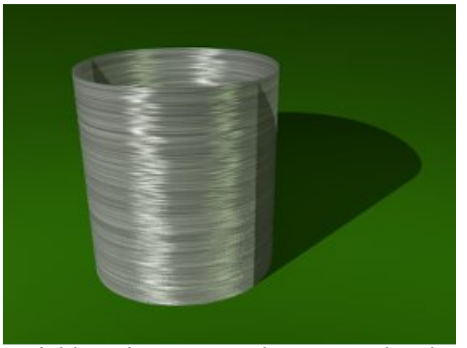
I go into side-view and unwrap the box as a cylinder from view.

Next I re-size the faces in the UV editor so that they will be similar to the face size of the 3d mesh. (Image 2)



We can now test out the first result. Turn on subsurf with a render value of 5, and create a material which has the Tangent V option, found inside the shaders panel, turned on. Set the objects' faces to smooth, and do a render preview. You will see a nice and smooth anisotropic highlight. (Image 3)

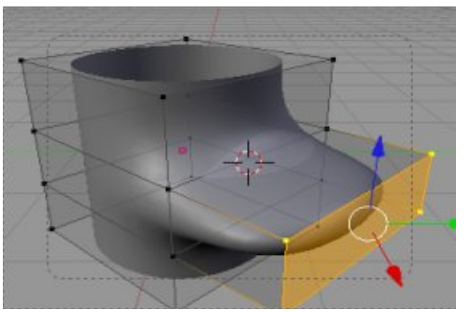
Maybe play a bit with the subsurf render value settings. Increasing and decreasing the value together with quick render previews will

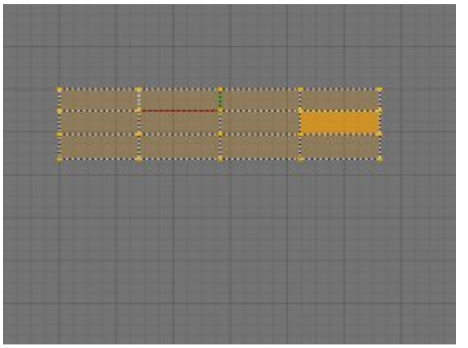


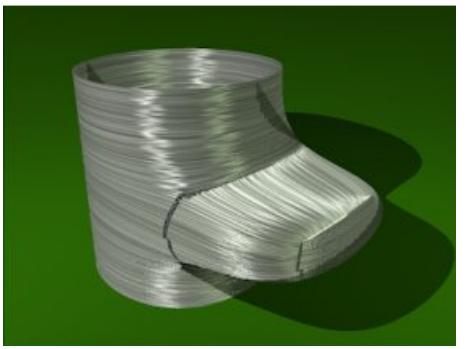
quickly show you how much the smoothness of the surface has an impact onto the smoothness of the flow of the highlight.

Step 2: Enhancing geometry and unwrapping:

Let's select one face and extrude it and review the result inside the UV editor. As you can see, the newly created faces do overlap and a quick preview rendering shows a bad result. (Image 4)
We need to select the newly created

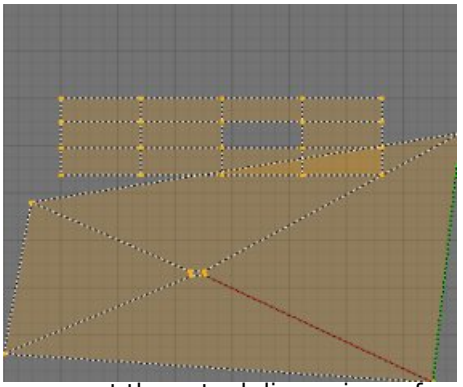




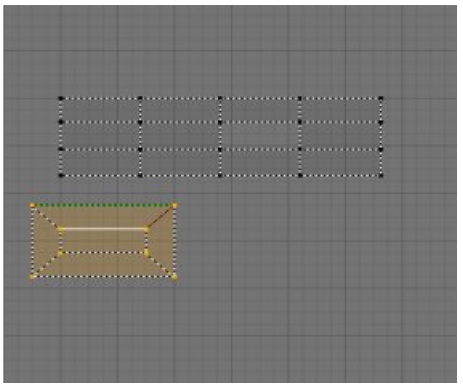


faces in the face selection mode and unwrap them with "U > Unwrap". Going back into the UV editor we will see that the scale proportions are totally off and we need to manually correct that. (image 5)

Select all faces of your objects in the Face selection mode, go into the UV editor and scale down the newly added faces so that they will

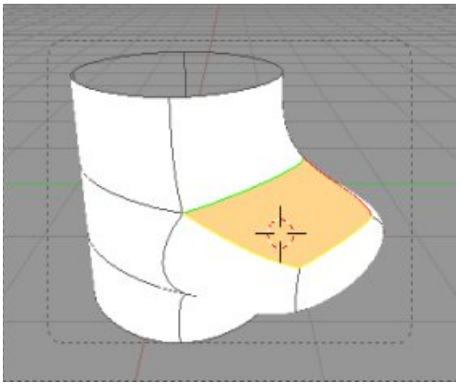


represent the actual dimensions of the 3D mesh. (Image 6)
It is very important to keep the face

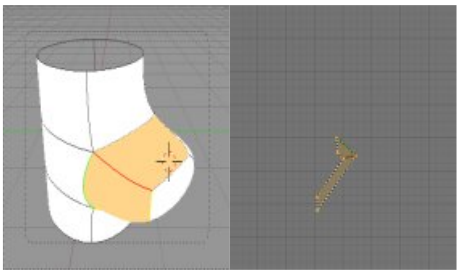


orientations intact. Go into the face selection mode and select on face.

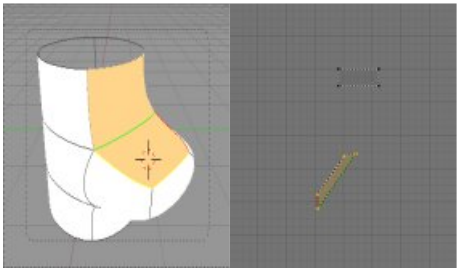
You will see two yellow, one green, and one red edge. Those red and green edges present the UV orientation. (Image 7)
Select a face next to that face and

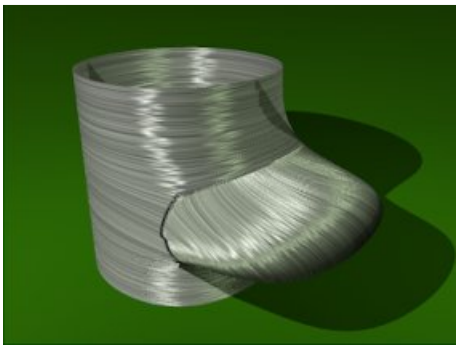


you will see that sometimes the color change depends on the orientation. However, in case the lower edge of one face is red and the top edge of the face below is green in the Face selection mode - and in the UV editor it is the same, you will be fine, they overlap and share the same coordinates in 2D space. (Image 8)



In case in the UV editor the two edges do not overlap the faces will have different UV orientations and positioning and the highlight will now flow through both connected faces. (Image 9)

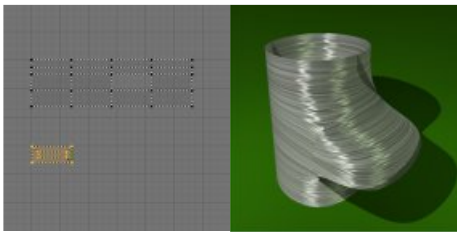




I also rotated the face to show how that will affect the orientation of the highlight and any textures using the uv coordinates.

Apply a scaled down voronoi texture and stretch it along one axis while deselecting the axis inside Map

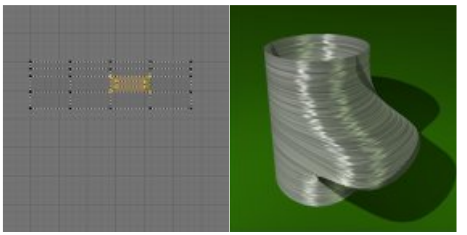
Input which is perpendicular to your anisotropic specular highlight direction. Map the texture to the Normal channel, select the UV mapping mode, do a quick render preview and enjoy the nice effect. (Image 10)



As you can see the highlight flows nicely over the geometry.

Step 3: Fixing texture flow:

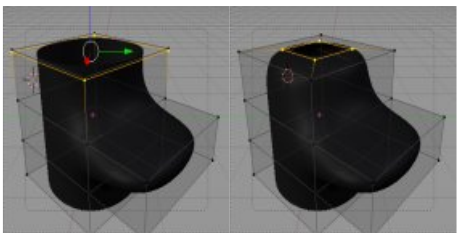
However the procedural texture seems to have some bad seams. This is because we did not weld the newly added faces inside the UV editor to the faces of the cube. We have to do this now. Select the faces of the new extruded parts and move them into the hole of the unwrapped cube. Using snap to pixels will help with moving the faces. Stretch them so that they fill out that hole and then selectively weld the overlapping face points together. Do a test rendering and you will see that the procedural texture perfectly flows over the geometry. (Image 11)

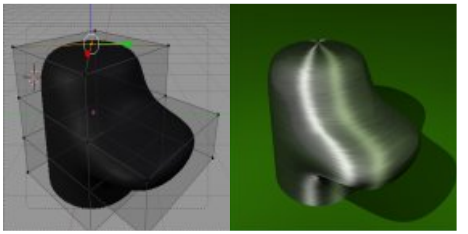


The drawback is that the last step distorts the proportional dimensions of the newly included faces inside the UV editor, thus distorting the texture a little bit. In other words, as you can see, the top sides of the extruded face shows stretched lines while the sides remaining have a finer grain.

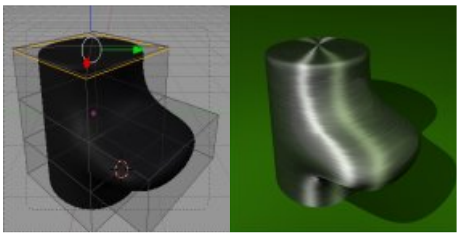
Step 4: Finishing up:

Let us close the top cap. Similar to the cylinder unwrapping, just add an edge loop cut very close to the top edge of the cube. Select the top edge and scale it down until the edges hit each other. (Image 12) To make them perfectly touch each other use the CTRL key and release the mouse button when the scale value reaches 0 for x,y, and z. (Image 13)

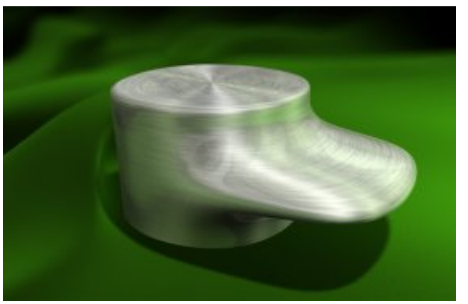




The last step is to include another edge loop cut close to the older top edge to create a nicely rounded edge for the pole cap. (Image 14)



(Finale Rendering: brushed metal)

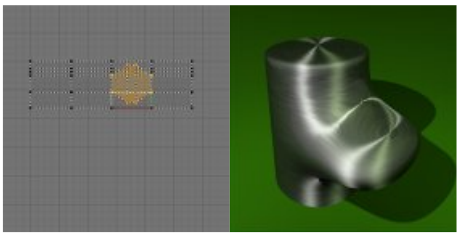


Comments

This overview should show you how to approach the texturing and modeling as well. Try not to make your object too complex. Also, keep in mind that with a very good texture you might not need to show all geometrical designs. Drawing quad based wireframes of your objects before modeling them will help you to layout the design and might save you a lot of time at the end. Starting over from the beginning is always a hassle - especially when you realize that with a bit more planing you would have already been done.

Through the overlaying of the second materials we also receive some nicely glowing specular reflections specifically along edges as seen at the top edge of the cylinder.

Experiment with the approaches I showed you. For example, the UV editor is quite cool to play with in case you want to create some funny anisotropic highlight distortions. Let faces overlap or rotate other faces and then welding them rotated again into the main mesh. The results might not be very realistic or useful for representing a real object. However they might be very useful for something experimental or other applications in which you can see this working. (Image 15)



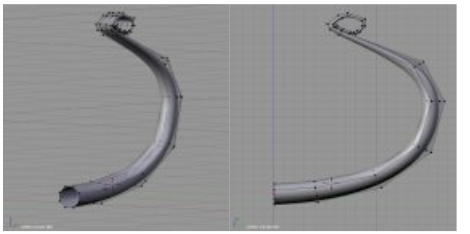
2. Hammered metal shader

Modeling of a bracelet

To create a forged bracelet I taper the ends with a rolling press and planish them round. *(Definition of planish: to smooth the surface irregularities in a sheet metal part with repeated hammer blows)* This produces nicely tapered round cone shaped ends which can be textured. For that I use mainly a ball shaped hammer. Through texturing I convert the round cross section into the shape of an octagon.

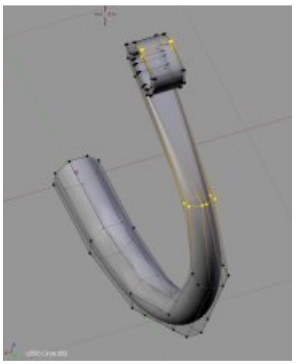
This shape has to be recreated through modeling. I start at the center of the bracelet and create an octagon. By extruding and evenly scaling down each new extrusion I can quickly build the main part of the bracelet. (Image 16)

Turning on subsurf, setting all faces to smooth, and creasing the edges

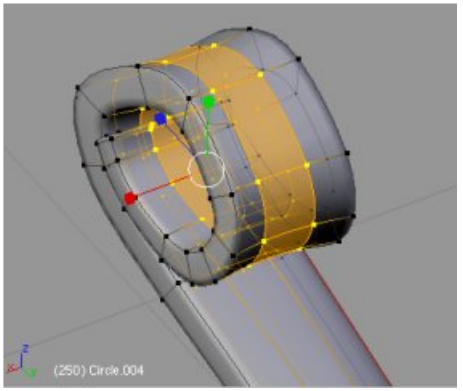


along the body will produce a nice and round body while the edges are sharp. I prefer to have a flat planished end which I roll into a loop. Because those ends are planished, there is hardly any texture left.

To model this I need to transform the octagon shape into a rectangular shape and move the center points to the sides in order to model the sharper edges. In the following steps I just gently scale down the thickness of the model and build a loop. (Image 17)



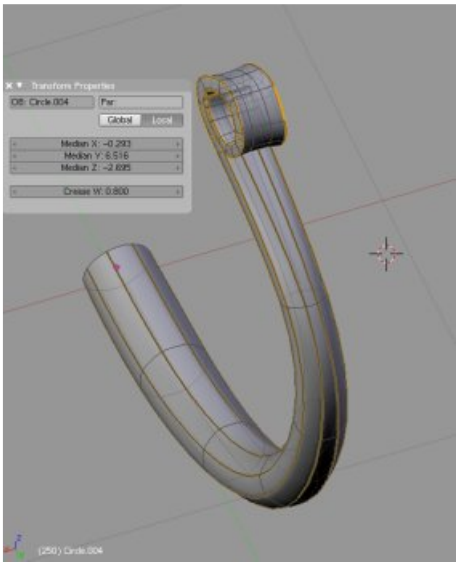
Because of the pressure applied during bending, the edges of the metal itself will roll up a little bit. In this case it might make more sense to leave the center points where they are and scaling down all those center points which define the loop. (Image 18)



We can use the edge crease option to selectively sharpen the edges of the bracelets. The edges along the loop have a high value to produce sharp edges, while the edges closer to the center of the bracelet have a lower value to produce a nice transition between the loop edges and the base edges. (Image 19)

Creating the shader

Before we get deeper into designing the shader I would like to present some basic information which is important to know to gain full control over your shader creation.



Displacement > how to control the displacement :
Every displacement map should utilize a gray scale. Use the 'No RGB' option inside the 'Map To' panel to turn color bands into gray scale information. Black and white are the two different ends of any displacement movement. The value of the displacement will control the distance between these two ends. Depending on if you set the displacement direction (ie. working outside towards the inside of the geometry), white will either move polygons outside or inside. Black would have the opposite effect. Neutral gray will not move any geometry.

So it is advisable that you do not work with white to black transitions. Only use Black to neutral gray transitions inside the color band and selecting the No RGB option. This way you can easily control the displacement direction with the 'Disp' button inside the 'Map To' tap.

Stencil > how to control the blending of displacement maps :
To mix two different textures together you have to position a masking texture between them which has the 'No RGB' and 'Stencil' options inside the 'Map To' panel selected. The 'No RGB' will give you better control over the blending because the blending is based on gray values. The 'No RGB' option also enables you to use the color band function. White means it will be 100% transparent and black means it will be opaque. Gray values will represent the transitional values between white and black, in other words - between transparent and opaque.

As mentioned, we have a transition from a textured octagon shaped body into an untextured rectangular body. Creating a material like this would be easy. You would only need to apply a cell texture as a displacement map for the main body, a second texture map with a much lower displacement value, and a blending texture to blend those two displacement textures together where the model changes its cross section shape.

To my knowledge it is not yet possible to mix different channels of materials together utilizing the displacement information. Currently only the color output is supported. Fixing this limitation will allow the creation of a layered shader with different displacement settings utilizing the node system.

A schematic view would look like this: Hammered metal shader tree: mixing

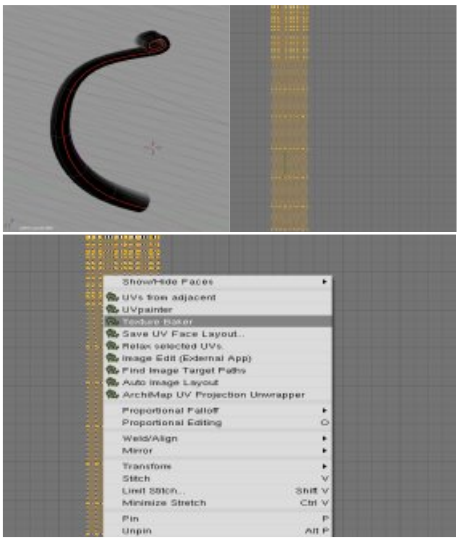
- channel one : 1. displacement texture (strong value voronoi)

- channel two : stencil texture (hand-painted uv map)

- channel three : 2. displacement texture (soft value voronoi)

This means that we will have to create a handmade displacement map which provides strong and soft displacement values where we need them. This is not very difficult and can produce very convincing results. With hand painted maps you also have more control over the end result. However they are also more time consuming to create.

Unwrapping and exporting the UV Face Layout
All we have to do is to unwrap our bracelet and correct the face layout inside the UV editor. Start the UV Face layout script and save it by hitting F1. (Image 20)



You can model one-half of the bracelet, and after texturing stitch a duplicate to the other end. The UV coordinates will be transferred to the duplication.

Preparing the texture baking
To bake a displacement texture we need to apply a texture to the Color channel inside the 'Map To' tap. As we did earlier, I recommend the voronoi texture. Inside the Map Input tap I increase the scale for x,y, and z to create small enough faces. Because I unwrapped the bracelet first the procedural texture will nicely flow along the cylindrical body.

Switch to the UV editor and start the Texture Baker script. The bigger the file size, the sharper the end-result will be. Because we work with pixels and not a procedural map, you should make sure that your displacement map will supply you with enough pixel information to prevent a blocky look.

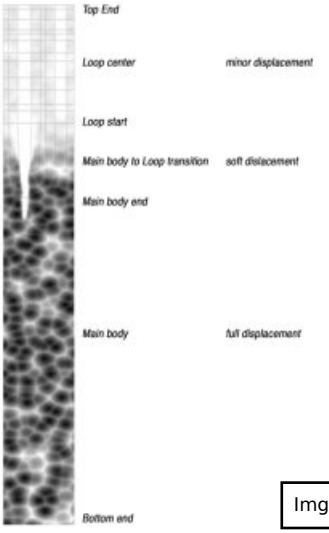
Composing the Displacement Map in Photoshop
Load both images, the baked texture and the UV face layout file into Photoshop and compose them together in one file. Apply a layer mask to the baked texture image

layer. Create an opaque to 80% transparent gradient, and fade the top part of the baked texture away starting at where the UV face layout map shows you where the loops of the bracelet start soon. Hide all other layers and export this image as a high res jpg file.

Creating a dirt map

Hammered silver surfaces tend to get dirty in deeper grooves because of tarnishing. The higher hills are often just polished through wearing the bracelet. Those dirty areas also have a lower reflection value than the high and polished areas. Save the newly created composition as a new photoshop file and call it something like reflection_dirt_map. Apply a level layer effect and increase the contrast between white and black. You can also lower the value output by moving the slide at the black side a bit away towards more a dark gray. This will keep strong white edges while brightening up the darker areas. Dirt is never uniform so apply a monochromatic noise filter to the baked texture layer. Hide all other layers and export this image as a high res jpg file. (Image 21)

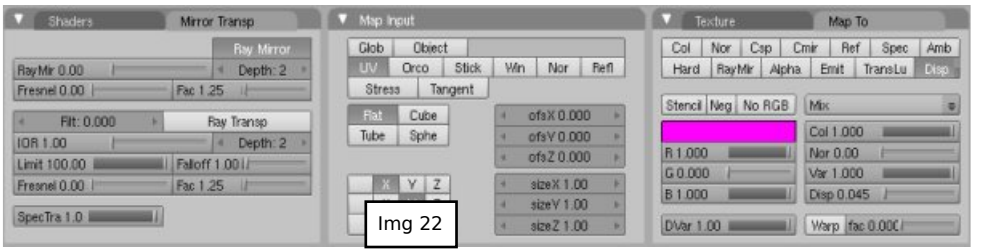
Creating the hammered silver material in Blender



Img 21

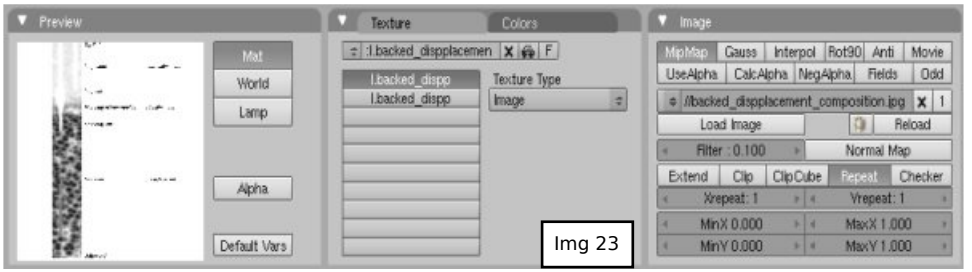
Switch to the Material window (F5) and create a material and turn Ray Mirror on. However leave the Ray Mir value untouched. Select Blinn for the specular highlights and create a sharp reflection. First we will create a channel for the displacement map. Go to the Texture tap and create a one channel. Inside the Map Input tap select UV go to the 'Map To' tap and deselect Col and select Disp. Set the Nor value to 0 and set the Disp value to something like 0.045. Depending on the gray scale of your displacement map you might need to increase or decrease the Disp value to achieve realistic values.

Secondly we will create a channel for the reflection and dirt map. Add a second channel and also set the mapping mode to UV. In addition to the Col option, select the RayMir option. Selecting the last one will make use of the values of the reflection map to define which area of the bracelet will be reflective and which will not. Gray values will be the transition between reflective and not reflective. Leaving the Col option selected will show the dirt values where the surface has no reflection value. Nice how both work together though one image map. (Image 22)



Img 22

Switch to the Texture window (F6) and create an Image texture. Load the baked displacement map jpg file. Deselect Interpolation and set the Filter value to 0.1 to prevent additional smoothing of your map. Create an additional Image texture and load the reflection dirt map jpg file. While you're at it, deselect interpolation and set the Filter value to 0.1 To make working with textures easier you should select the auto naming function, or give each texture a unique name. (Image 23)



Img 23

And we are done. You can hit render to preview you results. Check out the shadows. They show the displaced surface of the bracelet.

Comment
Hand created objects often show a level of imperfection or irregularity in surface and shape. As you can see the final model of the bracelet shows some of those deformations.


Hammered bracelet final render.

For example, each loop is different. I moved and removed some geometry. I did these steps after I merged both half shapes of the bracelet together. This way, I saved the time to create two custom sides first which also would have required me to create custom made image maps as well.

You should only do this after your are basically done with your model and texturing it. Furthermore, I did paint a bit into the displacement map to flatten out normally displaced areas. Through the process of bending the bracelet, I also planished out some areas of the textured surface. Using a hand painted displacement map gives you the ability to quickly show those irregularities. With a pure procedural map, you would not be able to do that so easily.

Displacement Modifier
The next version of Blender will have an interactive displacement preview. This displacement modifier will enable you to see the displaced geometry in realtime. The subdivision modifier level will affect the precision of the preview. The higher the level, the finer the detail, but it will be slower. This is great for smaller objects but not very fast to preview many at the same time. However, you tend to only work on one object at a time anyway.

Procedural textures or image maps are supported. Because Blender also comes with a built in paint module you can create your displacement image map in Photoshop, apply it in Blender to the displacement Modifier and preview the result. In case you need a little modification you can load the image

in the UV/Image editor and paint changes.

Switching between Object and Edit mode will force Blender to recalculate the displaced geometry and to show the changes to the image through the 3D mesh. With the possibly very high amount of polygons, it makes sense that Blender does not refresh any change automatically. Fortunately, his modifier comes with a very handy feature - It can bake the displaced geometry. This means that you can turn your preview into a solid new mesh and continue modeling with it. This enables the artist to use the displacement function, not only as a render time effect, but also as a solid modeling tool. With this tool, matching other objects to displaced geometries is finally not a nightmare!

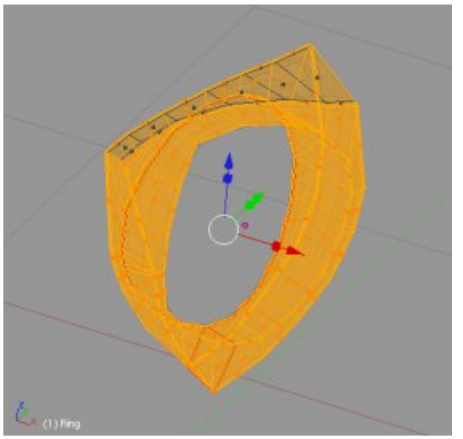
Warning
There is one down-side. Your scene file can quickly explode in file size. While my low polygon bracelet scene is around 1 MB a backed displacement geometry will add an additional 242 MB! This means that this function is great when you apply it to a small selection of objects without the highest subdivision level value.

3 Structured and darkened silver surfaces found with casted rings

Very common for casted silver jewelry which show a high level of details is that it is difficult to keep them clean. This causes a darkening of these areas which are hard to reach with a sulfur based chemical. Through wearing the ring all high points of the ring surface are going to be polished.

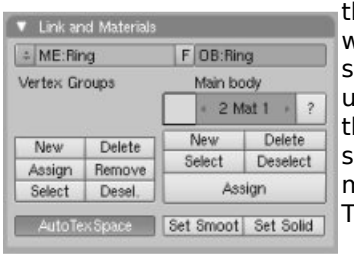
So what we would need is a material which provides a smooth polished and reflective surface for the main body and a material for the decorated area which shows a displaced surface, having a reflection value for the high points and a colored and non-reflective value to the low points of the displacement map. In reality those two different materials do not blend into each other. This means that we do not need to create any blending between them in Blender. To apply those two different materials to the ring we can simply create two different material indexes and apply them inside the Edit mode to different face selections of the ring geometry.

Select all of the faces which represent the main body of the ring and hit 'New' and then 'Assign' inside the Link And Material tap. Inside the modeling window press W and click Select Swap to select the other faces which will be displaced. Hit

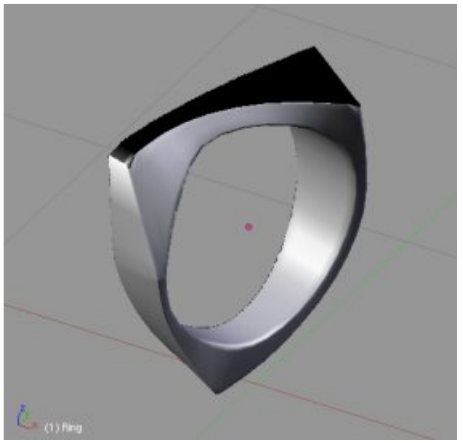


New and Assign as well. (Image 24)

You now create the two new materials, then assign those materials to two specific areas of the ring. Most ring designs can be modeled as a low polygon cage with quad faces that we will be smooth utilizing the subsurf modifier. Through



adding loop cuts we can pick out layout areas which will receive different textures. With some push and pulling you can quickly create some nice looking and comfortable to wear ring designs. (Image 25)
Time to create the materials. A



schematic view would look like this:

Structured metal shader tree:

- Main mesh faces: base material with silver settings

- Detail mesh faces: structured material with additional silver settings

- channel one : displacement texture

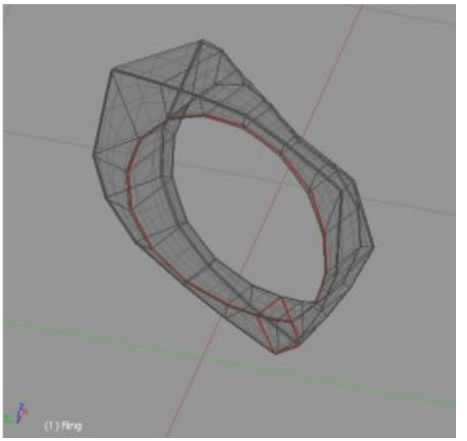
- channel two : inverted displacement texture

- channel three : color texture for darkened silver

We already created the material index for the ring. Switch to the Material window (F5) select the first material which represents the majority of the unstructured ring inside the Links and Pipeline tap. Rename that material to something meaningful, like "Main body". Set the Object color to black. To rebuild a perfectly polished silver surface, turn on Ray Mirror, set the RayMir value to 1.00 and create a sharp specular highlight with the Blinn shader.

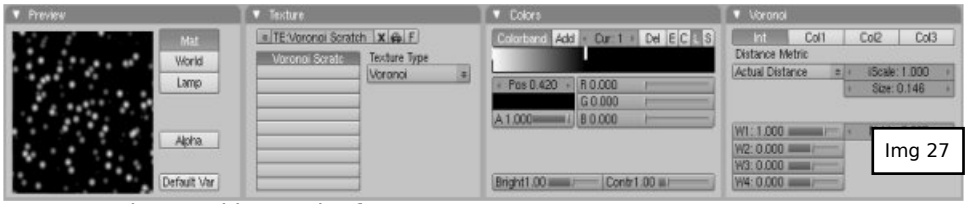
Even perfectly polished surfaces often still have some fine grooves and scratches. Through buffing you smooth out the sharp edges of those scratches and those produce a more even reflection which gives you the impression of a scratch free surface. An easy step to create that effect is to quickly unwrap the faces of the main body. Apply two loop edge seams along which Blender will cut the ring apart. (Image 26)

Switch to the Texture window (F6) and create voronoi texture. You can



scale it down a little bit and activate the Colorband option under the Colors tap. What we want is a bump map which only has few scratches here and there. For that set the black color Alpha value to 1.00 and turn cyan into pure white. We still have that cell structure grain typical for voronoi. Move the black color two thirds to the right side and the white color to the full left side. Know we have a nice black surface with some dots only. It might make sense to rename that texture something like "Voronoi Scratch". See Image 27.

Switch back to the Material Window (F5) and apply the newly create texture inside the 'Map To' tap only to the Nor channel and give it a very low Nor value. Inside the 'Map Input'



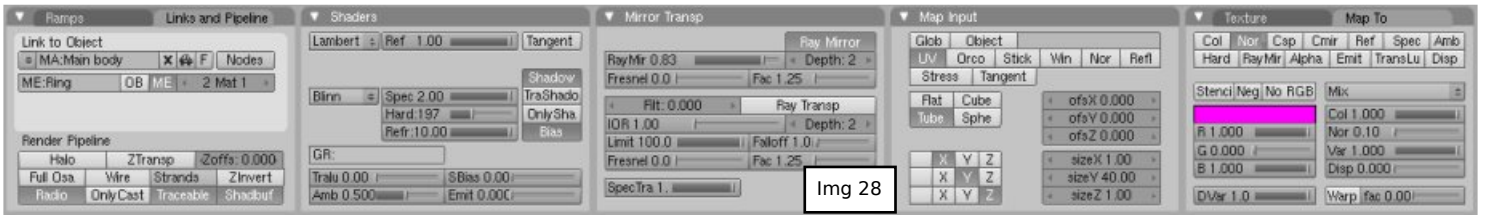
Img 27

tap, set the working axis for your model to a high value. This will squeeze the bump map to produce a brushed look. In my case I selected the Y axis and used a size Y value of 40. (Image 28)

the desired pattern size. Switch to the Colors tap and turn on Colorband. We only want the displacement go into one direction.

It should either rise away from the

After you made your decision it is time to create a blending mask. Select click into an empty channel inside the Texture tap and select the same Musgrave texture from the Add New menu. Click the 2 next to the small car icon to turn that instance of the texture into a single user. Go into the Colorband and move the black and medium gray slider a bit to the left. Add another color position and set that one to



Img 28

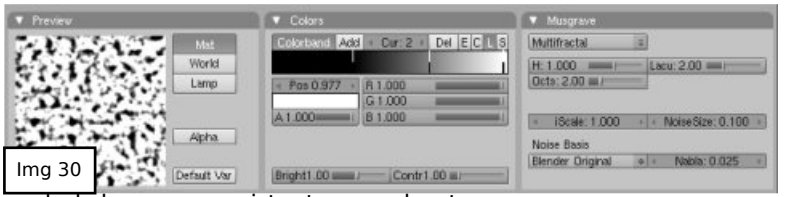
Now it is time to create the displacement material. Inside the Links and Pipelines tap, select the second material. Rename it to something like "displaced". Turn on Ray Mirror inside the Mirror Transparency tap but leave the RayMir value at 0. We will create a texture which will define the reflection values. You have to decide which procedural texture plug in you want to use for the displacement. In my case I selected a Musgrave. Play with the NoiseSize value inside the Musgrave tap to create

ring surface or it should sink into it. For that we need to set the colorband to a gradation between black or white to neutral gray. You can move the black slider away from one side closer to the gray point to define how soft the transition between untouched and displaced geometry will be. (Image 29)

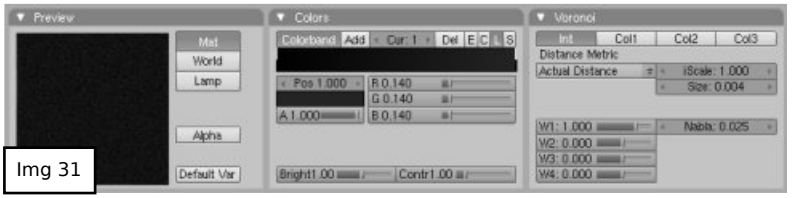
white and move this one a tiny bit away from the right side. You just created a mask which has slightly smaller holes than the displacement map through which the next texture will be visible through. (Image 30)

The last texture will be a dirt map. Select whatever procedural and image texture you want to use. I used a strongly

Img 29


Img 30

scaled down voronoi texture and set the colors inside the colorband to black and dark gray to produce a very fine and hardly noticeable grain look. (Image 31)
Go back to the Material window (F5) I stress again to make use of naming each texture to make managing


Img 31

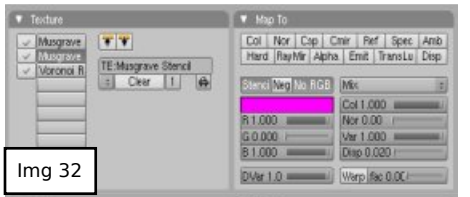
your channels much easier. Select the first channel with the displacement map. I set the Disp option to move faces away from the ring inside the Map To tap. Select the 'No RGB' button, set the Nor value to 0.00, and select a very light value for the Disp. I use 0.02 for example. (Image 32)

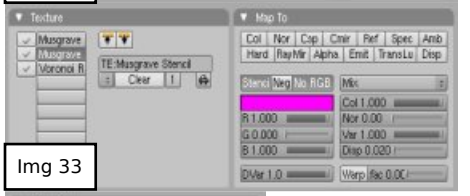
Select the second channel with the masking map. Deselect Col inside the 'Map To' tap and click 'Stencil' and 'No RGB'. Turning both on will make white transparent and will also enable the colorband. (Image 33)

Select the third channel with the dirt map. Double click RayMir. This will set the RayMir function to work on that part of the texture we did cut away through the masking map inside the second channel. This way the third channel will show the dirt map where we masking map did not key out and as well show the reflection. Furthermore does the texture we created also provide us with all needed RayMir value. (Image 34)

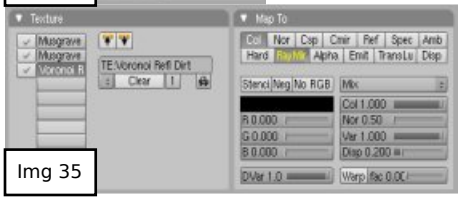
Having all those channels set up check out the Preview tap. Looks great doesn't it? (Image 35)

Inside the Editing (F9) window select the Subsurf option and give the render option a value of around 4 to 6. The higher the more precise your displacement will be during render time.. (Finale Rendering: Structured Ring)


Img 32


Img 33


Img 34


Img 35

Comment

When you work with perfect reflective surfaces it might make sense to give the different materials different base colors. This way you can see where you applied which material to which faces inside the 3D windows. I find that very convenient to work with. The higher the subsurf value,

the more precise the realtime preview.

When you work with reflections you always need to take the environment you put the ring into also into consideration. Your raytraced reflections have to reflect something. Quite often, raytraced objects do not work out for beginners because they only created the shader for the object, but did not continue to think about what those reflections will show. Putting the ring into the right setting will often drastically improve the quality of those reflections.

Soft irregularities inside reflections created through a second very blurred bump map will often increase the realism of reflections. There are hardly any perfectly flat surfaces in reality. Raytracing tends to produce those perfect visuals. As an artist you have the task to bring nature's irregularities back. Those fine details are often the key to producing realistic results.

Compositor based depth of field rendering

Small object studio photography makes high use of the depth of field effect to diffuse distracting background elements and to put focus on the main object. We can reproduce this effect with the compositor. There are two ways you could do it. The old way uses the well known Z-depth blur trick. In Blender you would use a pixel based matrix created through the z-depth information to blur the render output. The problem you would encounter is that pixels of the edges of the object in focus will be mixed together with the neighbor pixels, which belong to the background.

A real DoF effect would not use a Z-depth based image, but would render pixels depending on the distance to the camera in relation to the lens settings. The final bracelet rendering shows that problem. The top edges of the metal surface are blurred as well. However a way to prevent that is to divide the objects onto different layers and blur only the layer with the background objects. With using an Alpha Over node inside the compositor you can than mix the blurred background together with the object in focus.

You just have to be very careful with not blurring also parts of the background on which the camera focus sits on. Otherwise you might see a blurred surrounding while the focus

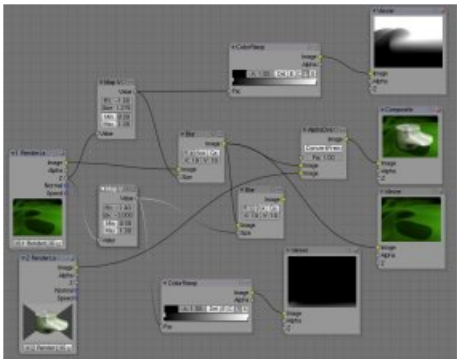


Image DOF

object is sharp. This would look physically wrong. In most cases you will only see a small amount of the foreground with object studio photography. This means that you can also ignore blurring the foreground.

The "Finale_Unwrapping" blend file contains all the required setups. Keep in mind that when you spread objects along render layers, you also have to make sure that all your lights also are active in those layers at the same time. Inside the Object window (F7) can select the different layers inside the Draw tap.
For more information about how to setup the render layers go to: http://www.blender.org/cms/Render_pipeline.747.0.html Look also into the other two jewelry blend files and

you will find similar blur problems.

Z-Depth sample based blurring

Recent changes in Blender brought to us by Alfredo de Greef brought us the new Defocus node for the compositor. From working with it, I got the impression that it seems to be a cross-over of the traditional Z-depth based blurring. It is not purely pixel matrix based, but rather uses samples like with a true DoF effect. The Z-buffer information seems to work as a helper in aiding Defocus to know where to sample. This will result in a sample mask which will only blur areas out of focus without bleeding into sharp edges of objects in focus. This is a enormous advantage over the old fake DoF trick because there are not artifacts and thus also no cleanup afterwards.
You can either use a traditional fStop like approach for blurring or you could use a Z-depth based image. If you go with the Z-depth image you can either use the Z information Blender provides you, or even supply the Defocus node with any other information of your choice.
One great advantage over the old Z-depth blur trick is that it's reasonably fast. Because it is a post production effect, you can use the preview function to quickly get a rough idea of the final effect. Depending on the amount of samples, the preview will be smoother or more grainy. For a final shot you only need to deselect preview and the compositor delivers the final result shortly after that.

Utilizing the Map Value node together with a Color Ramp node you can customize the z-depth image to make it fit your needs. The Map Value node sets the focal point while

Z-Depth blurring.

the Color Ramp node will adjust the size of the depth of field. And of course the nice Viewer node gives us a direct feedback on how the actually z-depth map looks like. And with the Zscale value you have control about the strength of bluring.

The fStop option is similar to what everybody knows from real cameras. A value of 128 will set everything sharp while 64 will

Custom Z-Depth defocus.

double the amount of blur according to Alfredo. This is for controlling blurring. Depending on your scene you might have to go down to values like and fStop of 4 to see the blur effect. To control the focal point the camera has a new option which is called DoFDist . You need to turn on show Limits for the camera to see the focal point. It is presented to the user as a yellow cross. Increasing the DoFDist will move the cross away from the camera.

This is a very comfortable and precise way on how to control the focal point and also knowing where it really is in contrast to the old fake

Defocus Fstop1 and Fstop2.

trick. But sometimes, depending on how you set up your scene and from where you point your camera, not everything seems to be in focus. The Defocus node comes with two tools which can be used to clean up artifacts but. Those are the Maxblur and BThreshold. However those tools you can also use to make the fo-

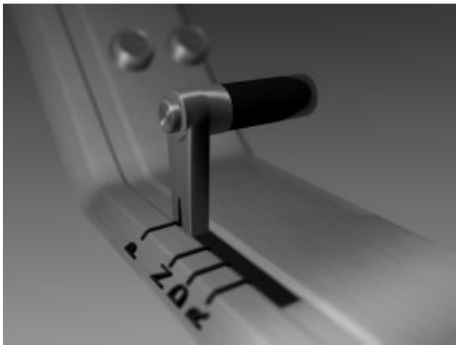
Defocus artifact1.

cus wide/longer while remaining the amount of blur as well.

Take a look at the Car_Interior.blend file. There you will see that the right part of the gear stick is out of focus.

Increasing the BThreshold will move the edges of the focal point more apart.

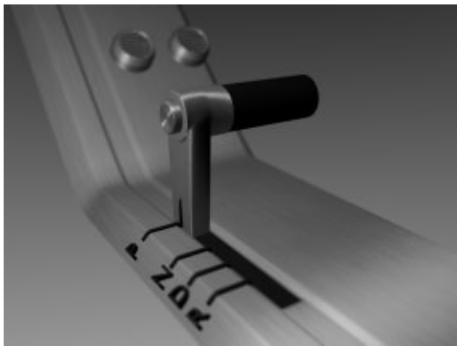
But as you can see there is also a blur artifact on the panel. The out of focus area starts too fast and there



Defocus artifact2.

is no transition between in focus and out of focus. You can fix this with using the Maxblur option. This will smooth out the edge.

As you can see the complete gear stick is in focus and the foreground



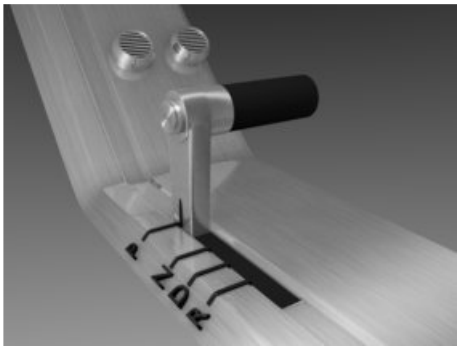
Car interior defocus fixed.

and background are nicely blurred. Because Defocus is a post production effect you can also just adjust the camera DoFDist value after rendering, and the compositor can recalculate the DoF effect again without re-rendering the complete scene.

For more information look here: http://blender.org/cms/Composite__Defocus.836.0.html

Car interior study

This image shows the types of geometries you might come in contact with. Cylinders, panels, and other basic shapes. Lucky us, those elements are easy to unwrap. Take a look at the scene files and how the individual objects are unwrapped to get an idea about how you should



Car interior

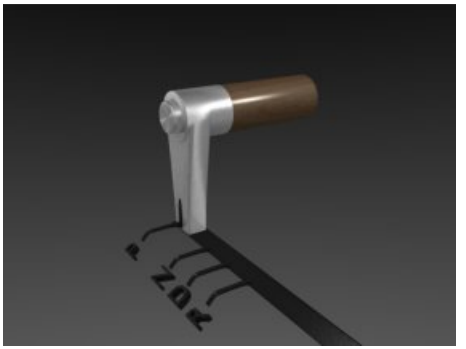
plan ahead you models for texturing. It can be a very big pain when you realize how difficult you made the object to unwrap.

The key to metal surfaces are smooth and flowing highlights and thus do you need continuous surfaces as well. Try not to make objects out of one mesh. It is helpful to study the construction and assembly of products before you try to model them. You will find out that many elements are made out of shells. Two shells can produce on container.

The gear stick panel is made out of one shell. I would rather try to model all elements separate. This way the UV-unwrapping and texturing will be much easier and more successful. Be aware of the relation between texture and object size. As you can see, the texture looks much more rough on the panel than the gear stick. Explore the scenes and you will see how simple the actual approaches are.

Looking at the gear stick in the UV-editor you can see how where I decided to apply the seam along which Blender cut the mesh apart. It is behind the stick and thus cannot be seen in the rendering. I used the unwrap function. Looking at the panel shows you how the downwards extrusion to make place for the gear stick to slide in is integrated. In this case no complex work was needed. The panel was unwrapped using the unwrap option. Looking at the air vents shows you that not everything needs to be perfectly unwrapped since you will not zoom in far enough to see the imperfection.

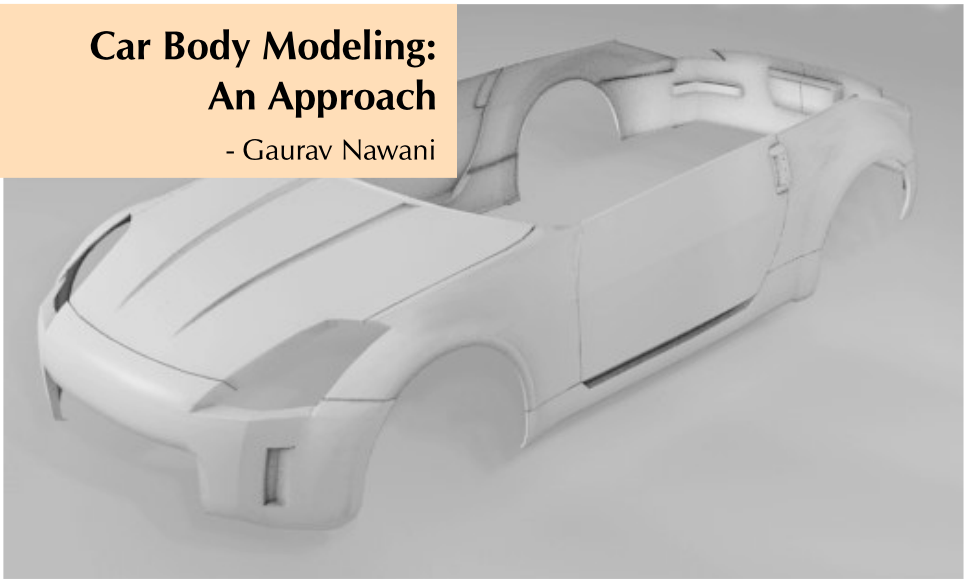
They were unwrapped with the unwrap function using a seam. The Car Interior 2 blend file includes the same scene just with modified shaders to show how sealed surfaces could be rendered. This is a quick setup explaining a possible approach. Take a look at how fine the brushed look could be and how it works together with the richness of the wood. The combination of different specular highlights adds additional readability for the viewer to understand the geometry and also the material properties.



Car interior2

Car Body Modeling: An Approach

- Gaurav Nawani



Level: Intermediate

Introduction

Unlike my previous tutorial on modeling the Toyota Celica car, this article is not a step-by-step modeling tutorial for a beginner. What I have tried to tackle in this article is some useful approaches towards modeling the car body in Blender. Although it looks and feels like a walk-through, I have tried to tackle some common problems while modeling a car in Blender which I hope will be useful to other car modelers.

Data Collection

It has been emphasized often, in many car modeling tutorials, the need to gather or research various data, including photographs of cars and its various parts. This provides the modeler with ready references during modeling.

Long before I start modeling, I try to remember the shape and unique features of the car body while going through the references and, if possible, the actual car. This helps me to consciously identify the possible problems in the mesh while I model.

Of course, having the reference images ready helps to solve any doubts that I may have during modeling.

Blueprints

Needless to say, blueprints are the very basis of accuracy that you want from your model. The more accurate they are, the greater the possibility of you being able to make a near-to-accurate model. I said near-to-accurate, because Blender does not offer measurement in real life units, even if you convert them to Blender units somehow there will always be a possibility of ambiguities in calculations. However, if you are close, they will be hardly noticeable.

While many users prefer to have the blueprints setup in a typical box-like fashion. I find them blocking my way while I model. So its mostly a personal preference. Choose whatever feels most comfortable to you. We have covered setting up blueprints in a box-style in one of our earlier issues, you can consult it if you like to use blueprints that way. I feel comfortable with using Blender's ability to show images in the viewports. For the viable scope of this article, I am assuming you know how to do that already.
There are various shape and sizes of

blueprints available on the web so, the first step is to edit them in an 2d image editor like GIMP or your preferred application. We are using the blueprints of the Nissan350z that can be found on the site www.the-blueprints.com.

There is often a few problems that most blenderheads have to deal with and those are:

- Different proportions in different views.
- Non-alignment of blueprints in different views.

Proportion problems occur as most users just cut the blueprints without taking into consideration that Blender's viewport somehow distorts the proportions depending upon the resolution.

Step 1 - Fixing the blueprint proportions problem

Make a single square document in your image editor with the maximum width being that of the blueprint. We choose a square as we are giving Blender images(views) of the same size, thus making sure that all are displayed in the Blender viewports with the same proportions.

First, copy all top, side and other blueprint views into separate layers in the image editor. While we are in the image editor, we can do the required transformations of different parts of the blueprint so that they can be represented in the correct viewport in Blender.

I always choose to have the top view of the blueprint in the vertical position as it compliments both front and back views nicely in the viewports. So, if your top view is horizontal you might want to make it vertical by rotating it.

Now create a new layer on top and draw a single-pixel line in the middle of the document. Taking this as reference, you can move the top, front and back blueprint views in the vertical middle. Now, again draw two reference lines, one above the car and the other at the bottom of the tires. This will give you the reference for fixing the position of the side, front and back views. If you did this step nicely, you should have the four different references setup like the images below.

Step 2 - Alignment problem in viewports

If you have followed the last part properly, you will have a good chance to set this up correctly. Now, all you have to do is split Blender's

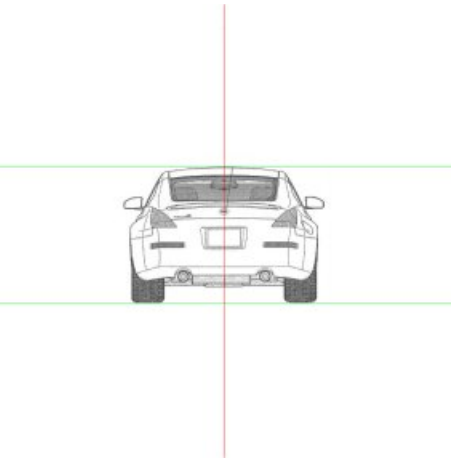


Illustration : Back View

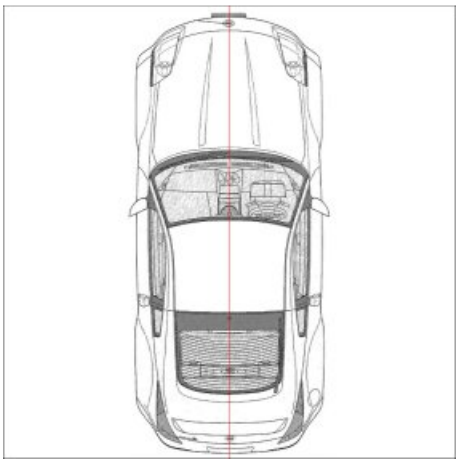


Illustration : Top View

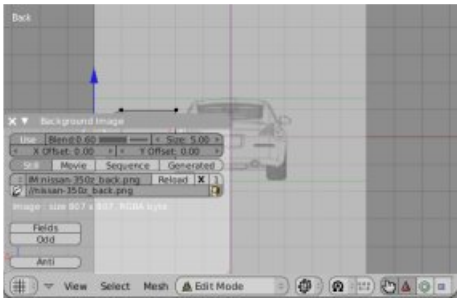


Fig 1: Image in the viewport background.

windows into four sections and include the respective image(views) in the viewport like the Fig 1. (The screenshot has been taken in Blender2.4 RC1).

You can enable the viewport name from the Preferences menu in the 'View & Controls' tab (View Name). So you can see which view projection you are in. Notice the Offset numerical buttons, these are the keys for you to align your image. Here, the reference lines we placed on the blueprint image in the last step will help you to get a good alignment.

Now just to make sure your alignment is working, all you have to do is insert a cube primitive and start extruding it in the top view. If the extrusion is outwards, then it should be outward in the front as well as the back view and towards the user or

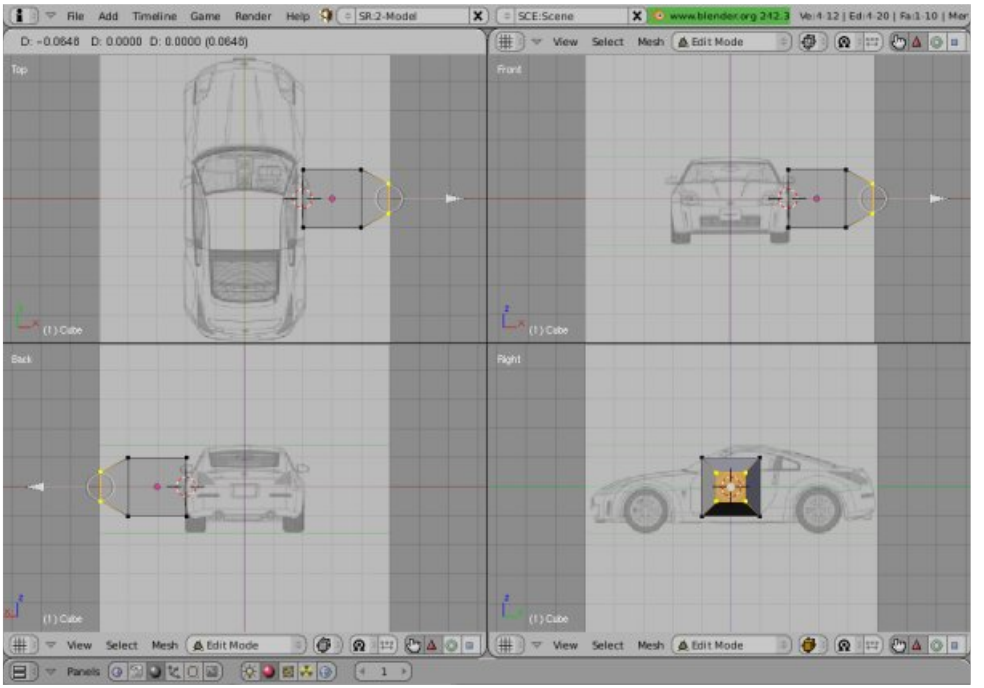


Fig 2: Checking the positioning of views.

the side view. See Fig 2 for example.

Modeling a Car's Body

There are two approaches towards polygon modeling a car in Blender. One is 'Box-modeling' and another is 'Plane-modeling'. They are completely opposite in nature. One attempts to create a shape in blocks with increasing amounts of detail, while the latter attempts to do it directly, but in parts.

Step 1 - Plane modeling

I have no experience with box-modeling so I will go about explaining a few approaches which could save you time and effort while modeling a car's body using 'Plane-modeling'. The best way to work is to work

with parts like the front fender or the bonnet, see illustration 3 for more.
We can start immediately by adding a plane in the side view. Before doing that, position the 3D Cursor as seen in Fig 2.

Illustration : Parts chosen for modeling.

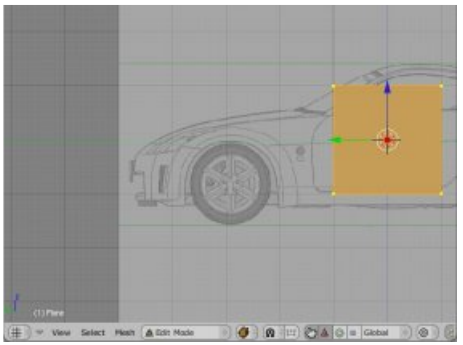


Fig 3: Adding a plane.

Position it over the front fender and reduce its size so that it can cover the contours of the fender just above the tire as in Fig 4.

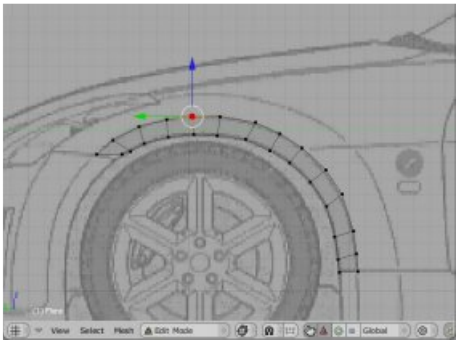


Fig 4: Plane modeling using extrusions.

Step 2 - Constructing geometry

You should enable Subsurf with a Level of 2, if your graphics card can handle it. One thing you need to keep in mind while you are modeling with blueprints is that you will have to model and recheck the mesh in three views. Three because once you are modeling, for example the front part, the back view will not be useful for any references. If you are not doing it often enough you will be ruining the geometry of the mesh. So, modifying then checking the geometry in the other views, as soon as you make some major changes in the mesh, will allow you to keep your mesh in a clean state.

Following this, now go to the top view and move the top vertices back a little and extrude a set of another vertices towards the bonnet of the car to cover the mesh as seen in the Fig 5a. Now, move the vertices as seen in Fig 5b, following the contours of the front fender.

TIP: While modeling a car body with SubSurf applied, it is advisable to initially use only very few extrusions to chalk out the basic outline of the shape or contour of the part. For example, in Fig 5a, we have used only three extrusions to make the side of the fender mesh. We will need more cuts in it to create more detail but, that can come later. This method

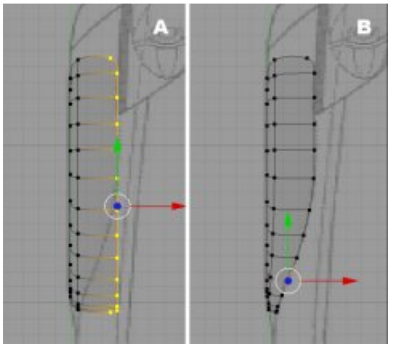


Fig 5: Extruding in top-view. 5b Adjusting.

will allow you to have greater control over the mesh's complexity.

Following this method, we will now immediately switch to the side view again and adjust the contour or shape as can be seen in Fig 6. The results can be seen in Fig 7 and Fig 8.

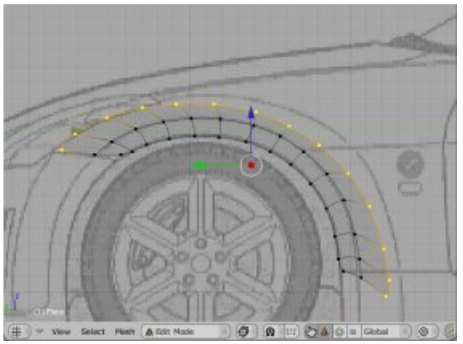


Fig 6: Adjusting shape in side view.

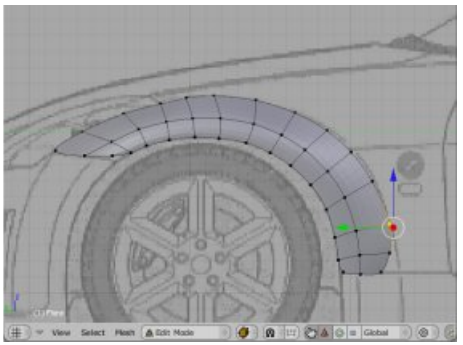


Fig 7: Shape after adjustments.

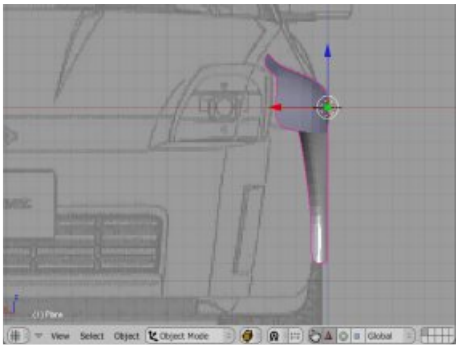


Fig 8: Adjust shape in the front-view.

Step 3 - Detailing

Always add more vertices or extrusions in which you can see the major portion of the part. For example, we can see that the side view shows a major portion of the fender. Switch to it and add more extrusions to cover other portions in this view. And following Step 2, correct the geometry in the top and front views. An initial and corrected version can be seen in Figs 9 and 10 respectively.

3a) Bends or contour lines

The shapes of a car often have pointed surface guides or bends like near the fender rim and a slight elevation near to the bonnet in this car (Nissan350z). These unique bends can be seen in the Fig 10 as highlighted orange edge lines.

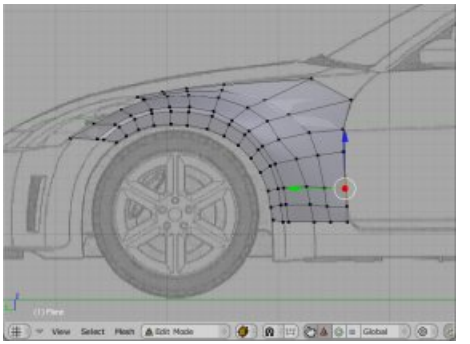


Fig 9: Covering fender with more divisions.

When using Subsurf, you will need approximately three similar edges, close together, to bring out sharp bends. In Fig 10, you can see that the fender rim looks pretty sharp as in the real life images of the car. To achieve the sharp blending, it will help if you keep your mesh in a clean state by only allowing it to

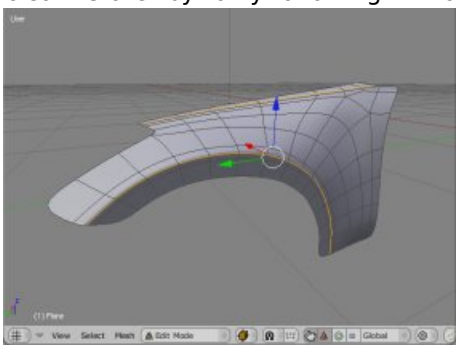


Fig 10: Emphasizing contours.

have quad faces near the place where the bend or sharp contour will be placed.

Placing a bend in the mesh is as easy as using the Loop Cut tool at the required place. Check Fig 11 for the results.

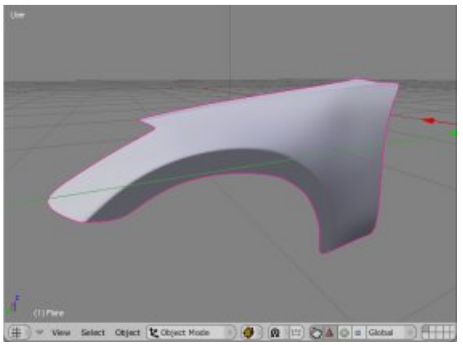


Fig 11: Resulting sharp bends.

3b) Clean edges

All the body parts of a real car are bent inwards to create a clean edge look. We, too, have to replicate this in our meshes to get quality models. Here, all of the outermost edges are selected and extruded once. Now the movement of the extrusion should be downwards for the parts that are at the top and, it should be inwards for the parts that are at the side and, again, inwards for the parts that are at the front and back. To better understand, study Fig 13 carefully. You will need to select and move the extruded parts separately.

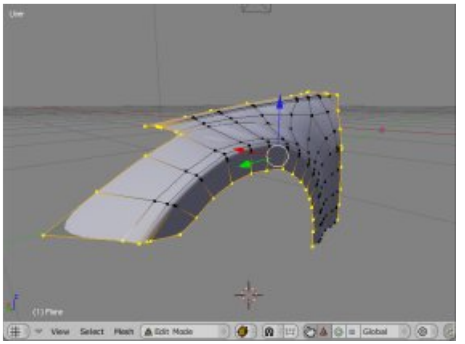


Fig 12: Selecting outer edges.

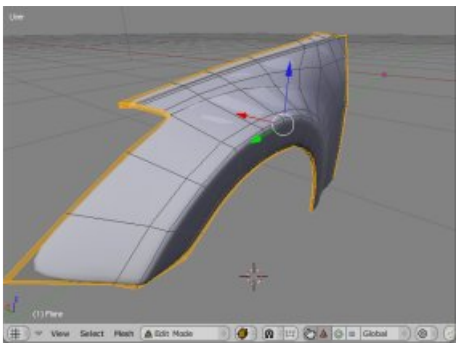


Fig 13: The edge details after extrusion.

3c) Clean corners

A car's body almost always has pretty sharp corners with very little roundness. For example, right now in the Fig 13 you can see the very front part corner is round in appearance. Again, utilizing the knowledge that with Subsurf, we can make it appear sharp by including edge loops near that corner. To get a better idea, we have switched to the top view in the Fig 14. Here, you can see the front corner before, and after, introducing an edge loop in Fig 14a and 14b respectively.

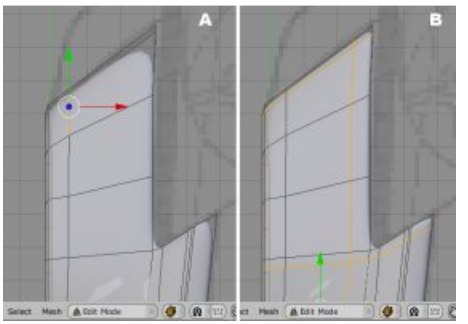


Fig 14: A Corners initially. B. Corners after adding edge-loops.

Notice that once you add two edge loops very near to each other (in a mesh with a large distance and difference of angle between two edge loops) you will get a deformation in the mesh. In the Fig 14b, the second edge loop will create a visible bulge on the top side if you will view it with perspective view. Here, you

have to manually move one of the edge loops to get a smooth surface. If it's safe, you can even merge the edge loops, for the most part, to reduce the edge complexity and retain the mesh's smoothness.

TIP: Snapping. It is a good idea to enable snapping from the Preferences menu. As car modeling is all about detailing, and when you move vertices of edges in perspective view, the snapping could help you a lot. For example, while extruding, to create and bend, it will give you nice known results.

3d) Smart parts

The heading may be a bit misleading but, it's a great way to reduce redundancy and also make sure that most of your meshes in the scene are of about the same mesh complexity. Carefully looking at Fig 15 will help you understand what is meant by "smart parts".

Actually, what we are doing here is simple. Once we finish one part of the car, we can use its corner edge as the starting point of another part. Here, we just duplicated and separated the edge, which was adjacent to the bonnet, and extruded it to create the bonnet mesh. You will notice that in Fig 15c all the vertices are aligned on the X-axis (red). This was done by pressing the S-key(scale) + X-key(restricts to X-axis) then, pressing Numpad-Zero and closing the scaling.

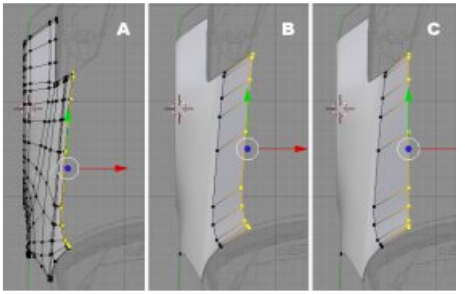


Fig 15: Starting the bonnet mesh from the edge of the fender mesh..

3e) Working with edge outlines

Sometimes a part will be a bit tricky to do by just extruding. As we can see in the Fig 16, the bonnet mesh is currently not smooth due to the fact that we have only done the extrusion in top-view.

If we switch to the side view and adjust the vertices, to get the required elevation, we have to deal with lots of vertices in the mesh, certainly not an easy situation. Instead of that, we selected the middle vertices and deleted them so that we get an outline of the left edge as can be seen in Fig 17.

Now, if we switch to side view we

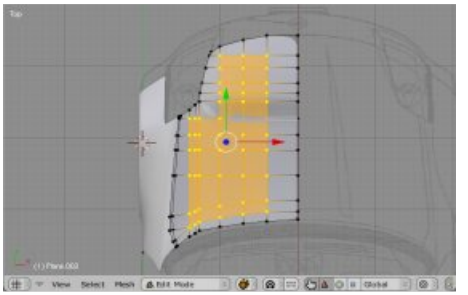


Fig 16: Deleting the inner vertices.

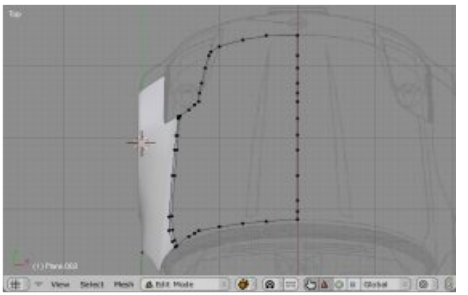


Fig 17: Edge outline left.

can easily position the vertices as in Fig 18; again we can repeat it in the front view as well, as in Fig 19. So now we have our outline set correctly. We are only left to extrude the edge (in the front view) from the center of the car towards the fender. Positioning their height in the front view and using only the number of extrusions as there were previous divisions. Remove the extra vertices as can be seen in Fig 20.

Remove double vertices (caused by extrusion over the old vertices) by selecting all vertices and using 'Remove Doubles' from the [W-key] menu in Edit mode. Also, remove those vertices that are not needed (seen as selected in Fig 20) to get

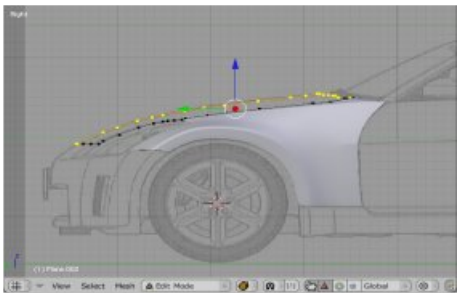


Fig 18: Positioning the vertices in side-view.

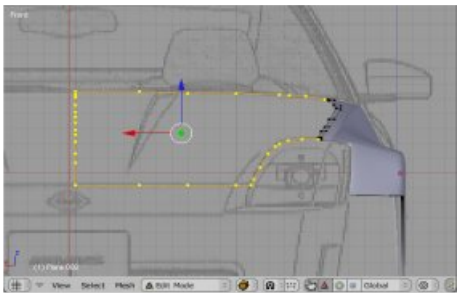


Fig 19: Positioning the vertices in front-view.

the smooth and proper curved surface of the bonnet in Fig21.

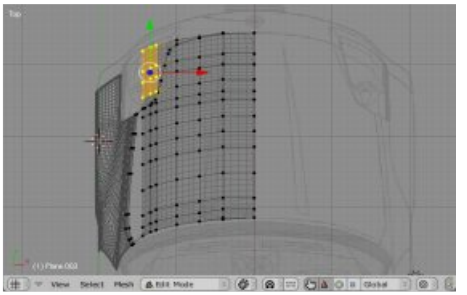


Fig 20: Removing the extra vertices.

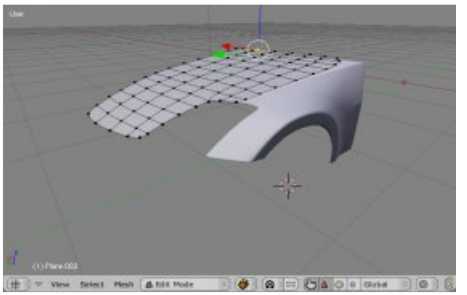


Fig 21: Properly curved bonnet mesh.

3f) MirroringSince almost all cars have a symmetric shape, our work is reduced to only modeling half of it then, mirroring the other half. Mirroring is pretty straightforward. If you ask Blender to mirror a part, Blender will take into consideration its pivot point and the origin, from when it was first made, as the point for mirroring. So, you always need to have the pivot point of the mesh in the center of the car.

Let's use the bonnet mesh again. Go to the top view and select the bonnet edge as in Fig 22. Position it correctly in the middle by referencing the blueprint below. Then, press [Shift + S] and select 'Cursor to Selection' in the popup. This will bring the cursor to the middle of the selection. Also, since this is the middle of the car too, the Mirror modifier will use this as the

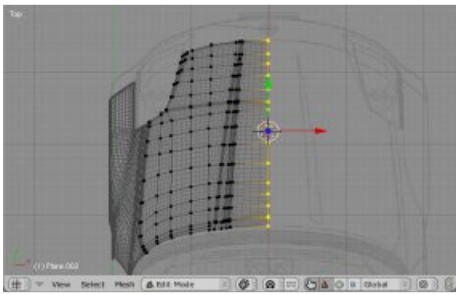


Fig 22: Positioning the cursor in the middle.

mirroring point. You will also need to position the mesh pivot at this point. To do that, get out of Edit mode and press F9, then click 'Center Cursor' in the Mesh Tab.

Now, apply the Mirror modifier and play with the axis setting to get the desired mirrored surface. The Mirror modifier also allows you to join the

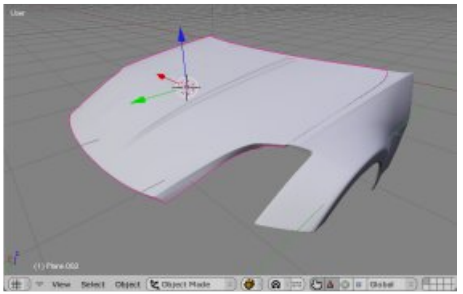


Fig 23: The mirrored bonnet mesh.

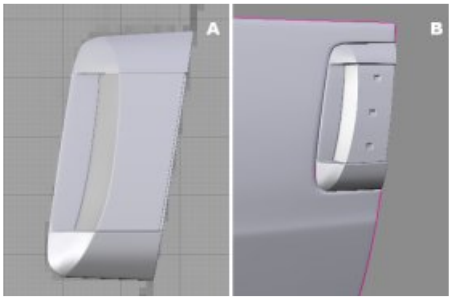
mesh and its mirror into an single mesh if you need. See the result in Fig23. Since you have already placed the cursor there, you can go ahead and select the front-fender mesh and make its pivot center at the current cursor position and add a Mirror modifier as well.

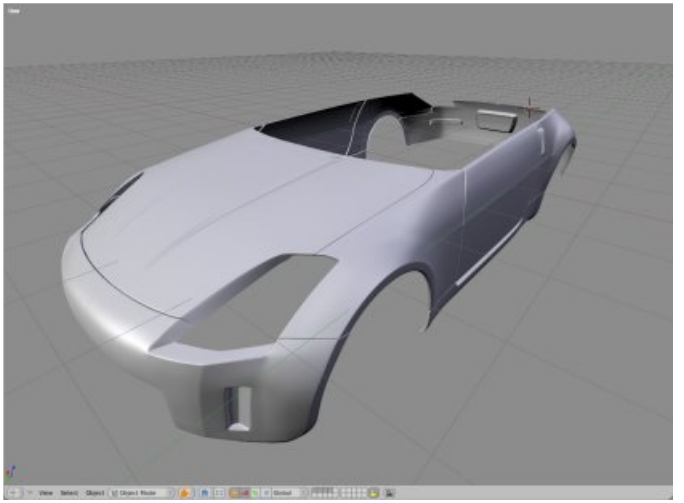
3g) Gaps to get natural looking body parts, make sure you always provide enough gap between the adjacent parts of the car body. You can observe this in real life on a real car. The distance between the bonnet and fender body will be in the range of 5-8mm. Doing that will also bring out the car body in renderings.

Conclusion

Well, that's all for a few interesting approaches in making the car body. I hope that it has been helpful to you. If you have something to share or feedback on these few steps, you are welcome to write to me at **gaurav@blenderart.org**. Further additions to this tutorial are also possible in future issues of the BlenderArt magazine. Some of the possible topics which can be covered are:

Modeling a realistic car lamp.
Modeling rims and tires.
Modeling the Interior.
Rendering the car.





Blender Basics

- Ronald Hess (harkyman)

Due to it's open and freely-shared nature, it's impossible to tell how many copies of Blender 2.42a are currently installed around the world, but it's a certainty that it has become more wildly popular than anyone had imagined when it was open sourced four years ago. Since then, how many times have the binaries been downloaded? Believe it or not, no one knows for sure, but ages ago (2004-2005) twelve month download totals approached 2,000,000. Without a doubt, more recent figures would make those look kind of silly.

But how many of those installations are tried for a week (or a day or an hour) then abandoned? Lots of us like using online tutorials, but there are just as many who feel more comfortable with a physical book on their desk. And so, with the Blender Summer of Documentation under its belt, the Foundation decided to tackle something even more ambitious: a book that would be suitable for 3D beginners, users coming from other 3D applications, and for more formal educational settings. Of course, more experienced users can benefit from the book too. How many of us have just flown by the seat of our pants regarding some aspect of our Blender work, be it materials, lighting or shape keying? This book can help you fill in the gaps. Of course, it also contains chapters on the new sculpt modeling tools as well as a great chapter on using the Compositor.

Blender Basics is a book that anyone interested in getting up to speed with some aspect of Blender, and 3D in general, can pick up and use, either cover to cover, or a la carte. It consists of a few chapters on the absolute basics: 3D concepts, the interface, and object manipulation. From there, readers can continue straight through the book if they choose, learning mesh editing, sculpting, materials, rigging and character animation, lighting, rendering, compositing and more. Or, if they are users integrating Blender into their existing 3D pipeline, they can pick and choose the chapters that most immediately fill their needs.
The book contains a bit of material from the Summer of Documentation, reworked to better suit a printed, modular format, and lots of original work by a great group of volunteers. To suit a variety of learning styles and the differing needs of the audience, each topic (like Materials, for instance) is divided into two sections: a tutorial part and a tools and practices discussion. For people who prefer to just jump right in, the tutorial sections provide immediate feedback and accomplishment, while getting them familiar with the tools in a practical setting. Those of us (like myself) who like to get an overview before working can read the tools and practices discussion, which lays out the most useful available options, as well as giving some hints and suggestions for a good work flow to follow.

In addition to the volunteer writers, I'm doing a fair amount of writing (about 35-40% of the book) as well as style-editing the whole thing. We have a bunch of technical editors lined up to make sure everything is accurate from a Blender standpoint, and are working to secure a professional proofreader to catch the typos and grammar bugs that might(!) slip by me.

Blender Basics will be available for pre-sale soon, and Ton and I are still working on the final scheduling for publication and distribution. Needless to say, book sales are one of the chief sources of funding for the Blender Foundation, and it's nice that such a thing can be a positive for everyone: the Foundation gets the revenue it needs to keep doing its amazing work, users get a great resource and a way to directly contribute to Blender's continued development, and the Blender community as a whole gets to once again show off its unparalleled strength with its great contribution to the content of the book.

I'm excited about this project, because I think that Blender Basics is going to be a book of superior quality and usefulness. I hope that you will, at least, consider adding it to your personal shopping list.

Ronald Hess

In addition to being a Blender artist, one of the (minor) blender devs, and the creator of BlenderPeople, Roland Hess (harkyman) earned his editorial keep as a writing major at the University of Pennsylvania in Philadelphia, enduring the brutal group workshops and one-on-one critiques of Penn's writing program. He writes far too many words a year at The Hess Report (hessreport.harkyman.com), Steel City Cowboy (steelcitycowboy.harkyman.com), and the BlenderPeople development blog (www.harkyman.com/bpblog).

OScar - Release 0.2

It is the goal of the OScar Project to develop a car according to Open Source principles. In our opinion, a car is not a vehicle full of high-tech gadgets. Instead, we are looking for a simple and functional concept to spread mobility. Form follows function.

Apart from that, OScar is not just a car. It is about new ways of mobility and the spreading of the Open Source idea in the real (physical) world.

At:
http://www.theoscarproject.org/
you will find a great community of developers and drivers who want to invent mobility anew and together.

The project started in 1999. In Dec. 2005, it reached release 0.2. Starting in 2006, everyone is welcome to participate.

Ciprian Ceteras (Cipix) Renaul Senic

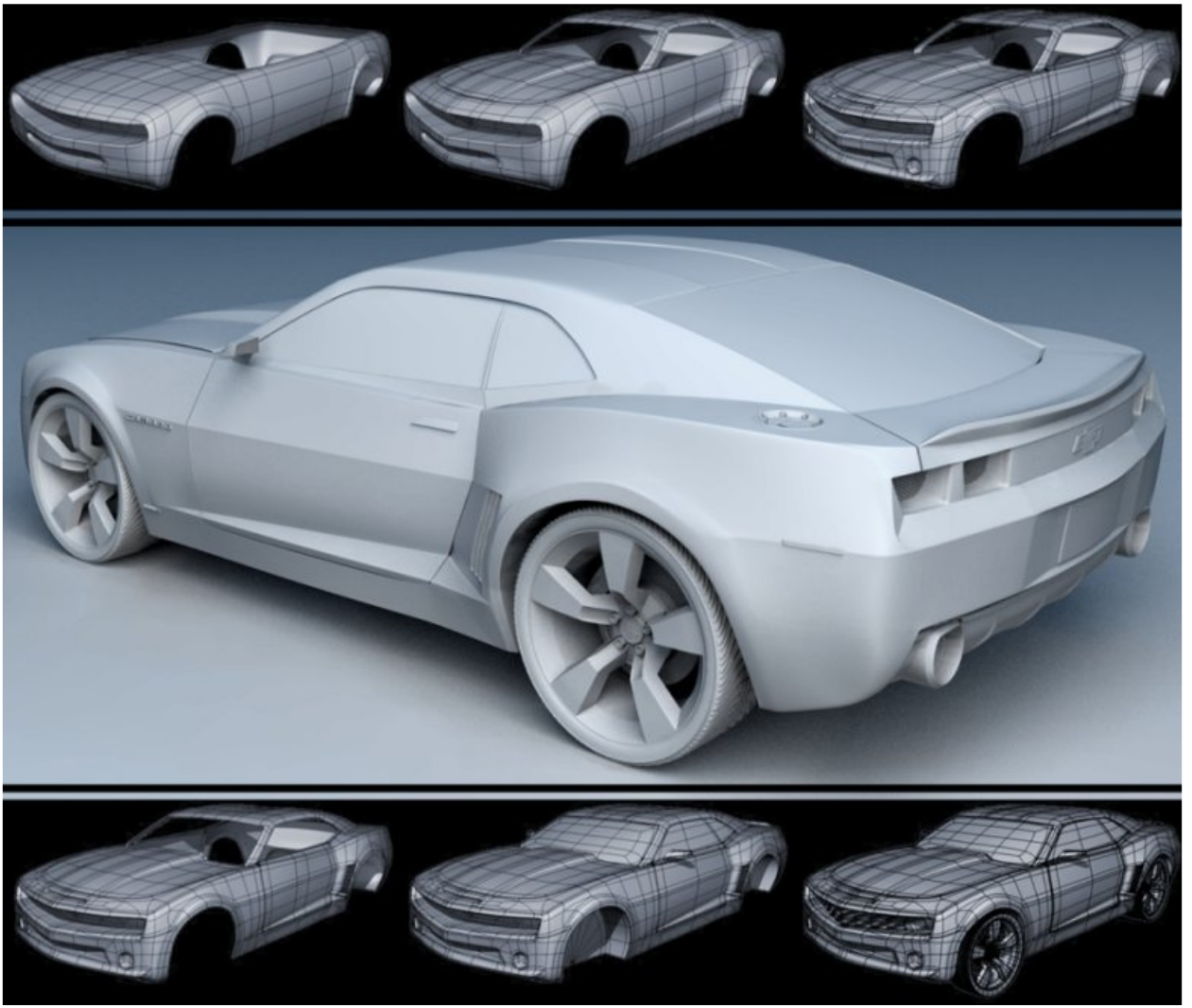
Ciprian Ceteras (Cipix) Renaul Senic

Jefferson Alves - F360 Modena

Simon Kindler - Discovery

Simon Kindler - Volvo XC90

Rishikesh Parkhe - Merc 300SL



Bruno Okano - Chevrolet Camaro (concept)

William Nunes - ToonCar

Zsolt Stefan - 'Cutting The Waves'

Blender World Cup #4

Call for sponsors!!

Many of you will be familiar with the blender world cup, which has been held for the past 3 years. Last year we received sponsorship from Respower (www.respower.com), a powerful online renderfarm supporting the .blend format for internal and yafray rendering.

This year however we have not officially received any sponsorship, and are hoping to make it the best year to date for prizes and participation. Commercial support is the primary goal for sponsorship, as the blender foundation can only help out so much.

So if you work for, or know a company that may be willing to donate, please contact doug@mud-puddle.co.nz in order to help make it fantastic for all in the community.

Blender World Cup - How it will happen this year

Where: www.blenderartists.org
When:
1st June - Qualifying entries opened
1st July - Final round opening

What:
The topic will be made up of 4 inspirations from below, the inspirations will be reduced to a set of 4 by the July the 1st, giving entrants the chance to think of possibilities before the final annoucement.

Era		*Location*
State		*Influence*
Beginning		Ground
Evolution		Industry
Historical		Sea
Growth		Religion
Recent	Era	Sky
Utopia		Aliens
Now		Space
War		Politics
Distant	Future	City
Desruction		Magic
End		Nature
Discovery		Heroism

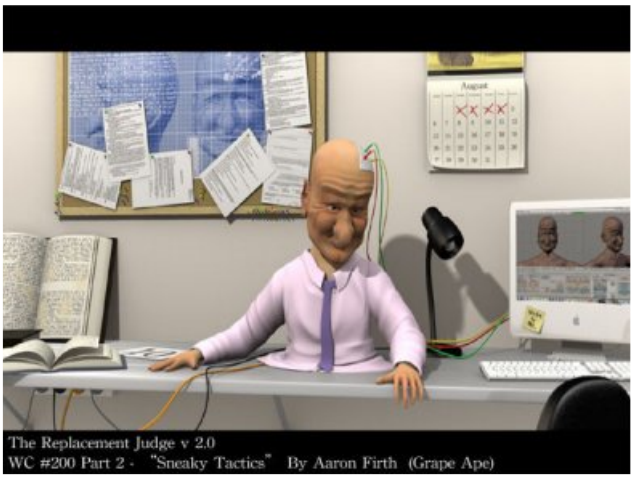
How:
As always we will have a qualifying round, however this year the round will allow anyone to submit their favourite existing entry to qualify for the final round. For Example entering Weekend Challenges will be a good way of qualifying

Prizes:
First second and third place winners will each receive an official etched blender world cup, as well as the prizes donated.

First

Second

Third



Blender World Cup - Previous Enteries.

Issue 9 March 2007

Theme: Outer Space

- **Modeling Aliens**
- **Modeling Spaceships**
- **Outer Space scenes and more . . .**

Disclaimer